



وزارة التعليم العالي والبحث العلمي
جامعة صلاح الدين

البرمجة بلغة البيسك وبعض تطبيقاتها في الرياضيات

تأليف

سعاده حسن علي البرزنجي

خوازيين سيدا فتاح

الطبعة الاولى

١٩٩٢

لتحميل أنواع الكتب راجع: (مُنْتَدَى إِقْرَأِ الثَّقَافِي)

پراي دانلود کتابهای مختلف مراجعه: (منتدی اقرا الثقافی)

پۆدابهزانانی چۆرهها کتیب: سهردانی: (مُنْتَدَى إِقْرَأِ الثَّقَافِي)

www.iqra.ahlamontada.com



www.iqra.ahlamontada.com

للكتيب (کوردی , عربي , فارسي)

البرمجة بلغة البيسك وبعض تطبيقاتها في الرياضيات

تأليف

سعاده حسن علي البرزنجي

خوازيين سيدا فتاح

قسم الرياضيات
كلية العلوم
جامعة صلاح الدين

بسم الله الرحمن الرحيم

المقدمة

في السنين الاخيرة اصبحت الحاسبات الالكترونية تحتل مكانة بارزة في المجالات المختلفة حيث انها تمثل سمة مميزة لعصرنا الحاضر ، فدخلت في المجالات العلمية ، الصناعية ، الاقتصادية والعسكرية نظرا لما توفرها للانسان من سرعة فائقة ودقة متناهية في انجاز العديد من العمليات الحسابية وقدرتها على حفظ واسترجاع المعلومات في الوقت المناسب لاتخاذ القرارات المناسبة .

ان التطور الكبير الذي شهدته الحاسبات الالكترونية ساهمت بشكل فعال على حل كثير من المعضلات التي كانت حلها سابقا وبدون الحاسبات الالكترونية امرا صعبا جدا ان لم يكن مستحيلا ومنها المعضلات الرياضية .

لقد رأينا من الضروري برجة بعض المسائل البسيطة و المعقدة التي تخص مواضيع مختلفة في الرياضيات للاستفادة منها وخاصة لطلبة الصفوف الاولى في اقسام الرياضيات ، الاحصاء والاقسام العلمية الاخرى في كليات العلوم ، التربية ، الادارة ، ...

يتضمن هذا الكتاب ثمانية فصول ، الفصل الاول يتناول دراسة الاجزاء الرئيسية للحاسبة واللغات المستخدمة والانظمة العددية المختلفة . اما الفصل الثاني فيتضمن الخوارزميات والمخططات الانسيابية وامثلة متنوعة .

وفي الفصل الثالث تم عرض اساسيات لغة البيسك والرموز المستخدمة بالاضافة الى شرح التعابير الحسابية والمنطقية .

خصص الفصل الرابع لعرض اهم الابعازات التنفيذية وغير التنفيذية في لغة البيسك . اما الفصل الخامس فيتناول شرحا مختصرا للمصفوفات وكيفية التعبير عنها بلغة البيسك ودورها في تبسيط كثير من المسائل الرياضية مع امثلة رياضية متنوعة .

الفصل السادس يناقش البرامج الفرعية بانواعها ومدى الاستفادة من هذه البرامج لتبسيط المسائل المعقدة او التى تتضمن تكرارات فى المسألة الواحدة .
اما الفصل السابع خصص لعرض مجموعة اخرى من الابعازات المهمة والاوامر المستخدمة فى لغة البيسك .
وفى الفصل الثامن والاخير تم وضع مجموعة مختلفة من البرامج المحلولة التى تتناول مسائل رياضية بحتة والتى تمت تنفيذ معظمها على الحاسبات الشخصية الوركاء .

البرمجة بلغة البيسك وبعض تطبيقاتها في الرياضيات

٥	المقدمة
١١	الفصل الاول : الحاسبة الالكترونية و اجزائها الرئيسية
١٣	1.1 ماهي الحاسبة الالكترونية
١٧	1.2 لغات البرمجة
١٨	1.2.1 لغات المستوى الواطي
١٩	1.2.2 لغات المستوى العالي
٢٠	1.3 الانظمة العددية
٢٠	1.3.1 النظام العشري
٢٢	1.3.2 النظام الثنائي
٣١	1.3.3 النظام الثماني
٣٤	1.3.4 النظام السادس عشر
	تمارين
٣٧	الفصل الثاني : الخوارزميات والمخططات الانسيابية
٣٩	2.1 الخوارزميات
٤٢	2.2 المخططات الانسيابية
	2.3 امثلة
٤٦	تمارين
٤٧	الفصل الثالث : لغة البيسك واساسيات لغة البيسك
٤٩	3.1 لغة البيسك
٤٩	3.2 اساسيات لغة البيسك
٤٩	3.2.1 الرموز
٥٠	3.2.2 الثوابت والمتغيرات

٥٢	العمليات والتعابير الحسابية	3.2.3
٥٤	التعابير المنطقية	3.2.4
٥٤	الايعازات في لغة البيسك تمارين	3.2.5

٥٧ الفصل الرابع : الايعازات في لغة البيسك

٥٩	ايعاز LET	4.1
٦٠	ايعازات READ - DATA , INPUT	4.2
٦٢	ايعاز PRINT	4.3
٦٤	ايعاز REM	4.4
٦٥	ايعاز END	4.5
٦٣	ايعازات السيطرة	4.6
٦٤	ايعاز GOTO	4.6.1
٦٦	ايعاز IF - THEN	4.6.2
٦٨	ايعاز ON - GOTO	4.6.3
٧٥	ايعاز FOR - NEXT	4.6.4
	امثلة	4.6.5

تمارين

٨٩ الفصل الخامس : المصفوفات

٨٥	مصفوفة ذات بعد واحد	5.1
٨٨	مصفوفة ذات بعدين	5.2
٩١	امثلة	5.3

٩٧ تمارين

٩٩ الفصل السادس : البرامج الفرعية

١٠١	البرامج المكتتية	6 . 1
١٠٤	الدوال الفرعية	6 . 2
١٠٨	الروتينيات الفرعية	6 . 3
١٠٩	امثلة	6 . 4
١١٣	تمارين	

١١٣ الفصل السابع : ايعازات اخرى والاوامر في لغة البيسك ومفاتيح التحكم ١١٥

١١٧	ايعازات اخرى في لغة البيسك	7 . 1
١١٧	ايعاز RESTORE	7 . 1 . 1
١١٧	ايعاز TAB	7 . 1 . 2
١٢١	ايعاز SPC	7 . 1 . 3
١٢١	ايعازات MID\$, RIGHT\$, LEFT\$	7 . 1 . 4
١٢٣	ايعاز LOCATE	7 . 1 . 5
١٢٤	الاوامر في لغة البيسك	7 . 2
١٢٤	الامر LIST	7 . 2 . 1
١٢٤	الامر RUN	7 . 2 . 2
١٢٥	الامر SAVE	7 . 2 . 3
١٢٥	الامر LOAD	7 . 2 . 4
١٢٥	الامر NEW	7 . 2 . 5
١٢٥	الامر DELETE	7 . 2 . 6
١٢٦	الامر KILL	7 . 2 . 7
١٢٨	تمارين	

١٢٩ الفصل الثامن : امثلة متنوعة في الرياضيات

الفصل الاول

الحاسبة الالكاترونية واجزائها الرئيسة

1. 1 ماهي الحاسبة الالكترونية

الحاسبة عبارة عن مجموعة من الاجهزة الالكترونية المختلفة ، يقوم كل جهاز باداء وظيفة خاصة وله القدرة على اتباع مجموعة من التعليمات (INSTRUCTIONS) لمعالجة الرموز (SYMBOLS) ، لذلك تعتبر الحاسبة اداة فعالة في حل المسائل وبسرعة فائقة .

ليست للحاسبة الالكترونية عقل تفكر به كما يعتقد البعض وانما هي مجرد آلة حيث انها لايمكنها من القيام باية عملية حسابية كانت ام مبرجة ما لم يكن هناك من يقدم اليها المعلومات والاورام ومن هنا تأتي ضرورة اتقان هذه التعليمات ليتم توجيه الجهاز بالشكل الصحيح لاداء المطلوب .

اجزاء الحاسبة الالكترونية THE COMPUTER DEVICES

تتكون الحاسبة الالكترونية من الوحدات الرئيسية التالية :

1 - وحدة الادخال INPUT DEVICES

تعتبر هذه الوحدة حلقة الوصل بين الانسان والجهاز ، يتم من خلال هذه الاجهزة ادخال المعلومات والبيانات والاورام الى الوحدات الاخرى للجهاز لمعالجتها .
توجد انواع مختلفة من اجهزة الادخال ولكن اكثرها انتشارا في الوقت الحاضر هي لوحة المفاتيح KEY BOARD . هي تشبه الالة الكاتبة (الطابعة) في تكوينها الخارجي .

2 - وحدة المعالجة المركزية CENTRAL PROCESSING UNIT

تعتبر هذه الوحدة الجزء الرئيسي في الحاسبة ، وهي الوحدة المسؤولة عن المراقبة والتحكم في سير المعلومات وتنفيذ العمليات الحسابية والمنطقية داخل الجهاز ، لذا فهي تعتبر دماغ الحاسبة الالكترونية

تكون هذه الوحدة من الوحدات الفرعية التالية : -

٢ - وحدة السيطرة CONTROL UNIT

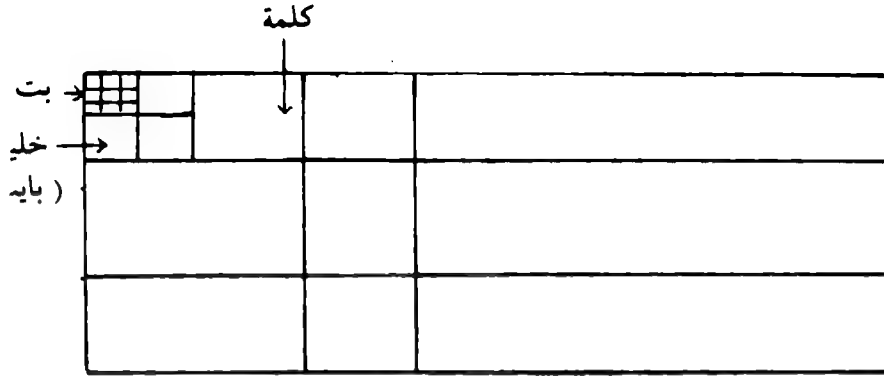
تقوم هذه الوحدة بالسيطرة والتحكم في تنفيذ التعليمات التي يتضمنها البرنامج عن طريق تفسير تعليمات البرنامج ومن ثم اصدار الاشارات اللازمة للوحدات المختلفة لتنفيذ هذه التعليمات وتتحكم في بقية مكونات الحاسب وتنسق بينها .

ب . وحدة الحساب والمنطق ARITHMETIC AND LOGIC UNIT

تقوم هذه الوحدة بتنفيذ العمليات الحسابية الجمع ، الطرح ، الضرب و القسمة ، وكذلك تقوم بمعالجة العمليات المنطقية التي يحتويها البرنامج .

ج - وحدة الخزن (الذاكرة) MEMORY UNIT

الذاكرة هي ذلك الجزء من الحاسبة الالكترونية التي تختص بتخزين البيانات والاوامر التي تدخل للجهاز ، وهي عبارة عن مخزن الكتروني مقسم الى وحدات خزن صغيرة تنظم فيه المعلومات بشكل يمكن الرجوع اليه بسهولة وبسرعة فائقة . يمكن تمثيل وحدة التخزين الاساسية ظاهريا وكأنها مستطيل مقسم الى مجموعة من المساحات وكل مساحة تسمى موضع LOCATION ، يخصص لاستقبال جزء معين من البيانات ويعطى لكل موضع رقم او اسم يسمى العنوان ADDRESS والموضع الواحد يمثل الوحدة الاساسية التي تقيس الذاكرة ويعرف باسم الكلمة (WORD) وكل كلمة مقسمة الى عدد متساو من الاجزاء وكل جزء يسمى خلية (CELL) ويختلف هذا العدد من جهاز الى اخر وتعرف هذه الوحدة باسم البايت (BYTE) وتتكون كل بايت من 8 خلايا تسمى بت (BIT)



وبذلك توجد وحدة قياس اكبر تسمى كيلوبايت KELOBYTE حيث ان

$$1 \text{ KELOBYTE} = 1024 \text{ BYTE}$$

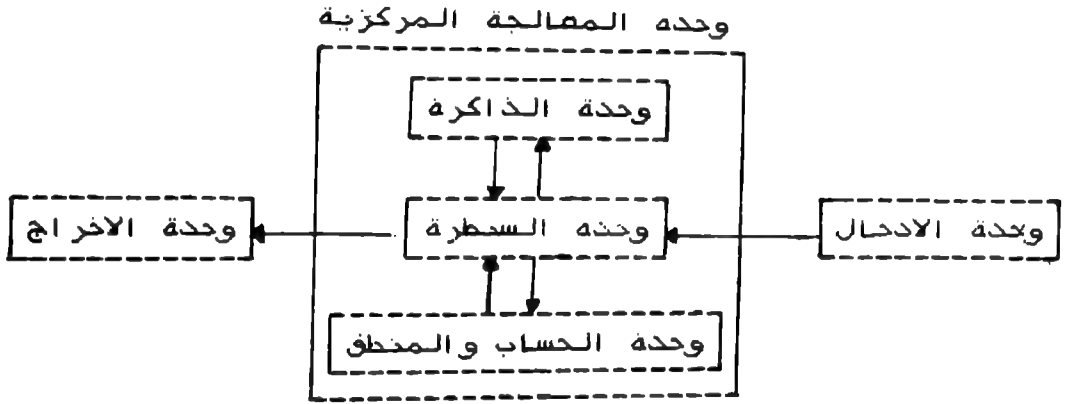
فاذا قلنا ان سعة الحاسبة تساوي (28K) فهذه تعني ان ذاكرتها تسع $28 * 1024$ بايت من المعلومات وتختلف سعة الحاسبة من جهاز الى جهاز اخر .
 بصورة عامة تتكون الذاكرة في معظم الحاسبات الدقيقة من جزئين رئيسيين احدهما يسمى ذاكرة القراءة فقط (READ ONLY MEMORY) او (ROM)
 يسمى الثاني ذاكرة الوصول العشوائي (RANDOM ACCESS MEMORY)
 او (RAM) يستخدم الاول لحزن برامج التشغيل ، ويستخدم الثاني لحزن البرامج العامة .

لايمكن استخدام الذاكرة الرئيسية للحزن الطويل للبيانات او البرامج وذلك لان الذاكرة الرئيسية سعتها محدودة بالاضافة الى ان ما يحزن في الذاكرة من بيانات وبرامج معرضة للضياع بسبب انقطاع التيار الكهربائي وبعض المؤثرات الفيزيائية او اسباب اخرى ، لذا فان من الضروري استخدام اوساط اخرى لحزن البيانات والبرامج بعد نقلها من الذاكرة وتعاد اليها عند الحاجة ، تسمى هذه الاوساط بالذاكرة المساعدة (AUXILLARY MEMORY) ، ومن هذه الاوساط الاقراص ، الاسطوانات ، الاشرطة المغنطة . الخ .

3- وحدة الاخراج OUTPUT UNIT

تقوم هذه الوحدة باستخراج المعلومات والنتائج التي حصت عليها الحاسبة بعد تنفيذ البرنامج الى الوسط الخارجي . توجد انواع مختلفة من اجهزة الاخراج ولكن اكثرها استخداما هي الطابعة الخطية LINE PRINTER .

يمكن توضيح الوحدات الرئيسية للحاسبة وكيفية ترابط عمل كل وحدة بالآخرى بالشكل التالي :-



اي ان عمل الحاسبة الالكترونية يتم على مراحل :-

- 1 - استقبال المعلومات عن طريق وحدة الادخال .
- 2 - تخزين هذه المعلومات في الذاكرة .
- 3 - استخراج المعلومات من الذاكرة عند الحاجة .
- 4 - اجراء العمليات الحسابية واتخاذ القرارات المنطقية .
- 5 - استخراج النتائج الى الوسط الخارجي عن طريق وحدة الاخراج .

التعامل مع الحاسبة الالكترونية

يتركز التعامل مع الحاسبة الالكترونية بصورة عامة في شطرين اساسين يمثلان تكاملا لاداء وتنفيذ العمليات الحسابية والمنطقية ، وهما : -

1- البرمجيات SOFTWARE

عبارة عن دراسة تفاصيل المسألة المراد معالجتها ووضع البرامج التي تؤدي الى حل المسألة . وبأختصار فان البراهجيات هي البرامج واللغات المستخدمة في كتابة البرنامج .

HARD WARE 2. المعدات

تمثل الاجزاء الداخلة في تركيب وعمل الجهاز ، ومن اهم هذه المعدات الوحدات التي تتكون منها الحاسبة الالكترونية .

PROGRAMING LANGUAGES

ان لغات البرمجة هي وسيلة تعامل الانسان مع الحاسبة الالكترونية لذا يجب ان تكون هذه اللغات مفهومة للطرفين الانسان والجهاز تتكوّن لغة البرمجة من مجموعة من المفردات اللغوية التي تمثل شفرة (CODE) معينة يفهمها الجهاز ، ويتم التعبير عن هذه الشفرة اما على شكل ارقام تسمى شفرة رقمية او على شكل رموز وتسمى شفرة رمزية او على شكل كلمات محفوظة داخل الجهاز وتسمى كلمات محفوظة (RESERVED WORD) .

باستخدام هذه الرموز او الارقام او الكلمات يمكن للحاسبة الالكترونية تركيب مجموعة من التعليمات او الاوامر التي يمكن الجهاز فهمها وترجمتها وبالتالي تنفيذها .
مجموعة التعليمات والاورامر المتسلسلة منطقيا لحل مسألة ماتسمى البرنامج .
يمكن تقسيم لغات الحاسبة الى :

1. 2. 1 لغات المستوى الوطني LOW LEVEL LANGUAGES

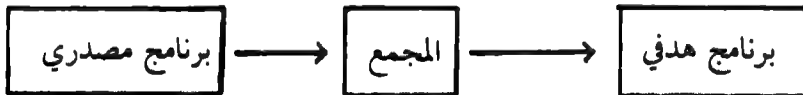
وتتضمن:

١ - لغة الآلة MACHINE LANGUAGE

وهي اللغة الوحيدة التي يفهمها الجهاز ويستخدمها للاتصال بين وحداته المختلفة ، تختلف هذه اللغة من حاسبة إلى أخرى وذلك حسب تركيبها الداخلي حيث أن لكل جهاز لغة خاصة به لتنفيذ الأوامر الصادرة إليه حسب نظام معين ، مثلاً في النظام الثنائي فإن لغة الآلة تعتمد على شفرة رقمية (NUMERIC CODE) تتكون من الأرقام 0 , 1 (صفر ، واحد) أي أن كل الكلمات والإرشادات والرموز تتم ترجمتها إلى الأرقام 0 , 1 .

ب - لغة التجميع

تستخدم هذه اللغة شفرات رمزية بدلاً من الشفرات الرقمية وتختلف هذه الشفرات من جهاز إلى آخر وذلك حسب نوع وتصميم الجهاز . البرنامج المكتوب بهذه اللغة يعرف باسم البرنامج المصدري (SOURCE PROG) والذي يحتاج إلى الترجمة إلى برنامج بلغة الآلة والذي يعرف باسم البرنامج الهدفي - OBJEC- TIVE PROG ليتم تنفيذه ، وتتم عملية الترجمة من خلال برنامج خاص يعرف بالمجمع (ASSEMBLER) .



ن أترجم هذه اللغات تحتاج الى مهارات وخبرات ومعرفة بالجهاز المستخدم مما يشكل صعوبة في تعلم هذه اللغات ، لذا كان من الضروري إيجاد لغة او لغات معينة تصلح مع معظم الاجهزة ، ومن هذه اللغات لغات المستوي العالي .

1 . 2 . 2 اللغات ذات المستوي العالي

HIGH LEVEL LANGUAGES

لتلافي صعوبة البرمجة باللغات الواطئة لغير الراغبين بالالمام بالتفاصيل الدقيقة للجهاز لقد تم وضع لغات خاصة لتصميم البرامج بشكل بسيط وغير مرتبط بالتركيب الداخلي للجهاز ، بحيث يستطيع المبرمج برمجة المسألة بكل سهولة بعد معرفته للامكانيات المتوفرة للجهاز وكذلك الالمام بقواعد واساسيات البرمجة بهذه اللغات .

يحتاج البرنامج المكتوب بهذه اللغات الى ترجمته الى برنامج بلغة الالة من خلال برنامج خاص يعرف بالترجم (COMPILER) .



اي تتم ترجمة البرنامج المصدري الى الهدي من خلال المترجم ، وتتم عملية الترجمة هذه من قبل الجهاز نفسه .

لقد تعددت اللغات ذات المستوي العالي نتيجة لتعدد طبيعة المسائل والتطبيقات التي يمكن استخدام الحاسب في حلها ، لقد ظهرت لغات للتطبيقات العلمية واخرى للمسائل التجارية واخرى لمعالجة كميات كبيرة من البيانات الخ . . .

ومن أبرز هذه اللغات :

- 1 - لغة الفورتران (FORTRAN LANG.): تستخدم هذه اللغة لحل المسائل العلمية والهندسية .
- 2 - لغة الكوبول (COBOL LANG.): تستخدم في التطبيقات التجارية .
- 3 - لغة الكول (ALGOL LANG.): تستخدم في التطبيقات العلمية .
- 4 - لغة البيسك (BASIC LANG.): وهي اللغة البسيطة والمستخدمه للاغراض العامة .

1.3 الأنظمة العددية

NUMERIC SYSTEMS

ان كتابة البرنامج بلغات الحاسبة تحتاج الى دراسة الانظمة العددية المختلفة .
والواقع ان لغات المستوي اللعالي تتطلب منا معرفة النظام العشري المتعارف عليه في حياتنا اليومية ، اما الانظمة الاخرى فهي غير اساسية بالنسبة لها والتي تعتبر لغة البيسك واحدة منها .
اما لغات المستوي الواطي فانها تستخدم انظمة متعددة وحسب نوع الجهاز .
فيما يلي مجموعة من الانظمة المستخدمة :

1.3.1 النظام العشري

DECIMAL SYSTEM

يسمى هذا النظام احيانا بالنظام الموسع ويتكون من عشرة رموز مختلفة وهي 0, 1, 2, ..., 9 وان الاساس لهذا النظام هو العدد (10) (يعتمد عدد رموز اي نظام على اساس النظام) ، تكتب الاعداد في هذا النظام باستخدام هذه الرموز استنادا الى تركيب معين وكما يلي :
العدد الصحيح يتكون من عدد من المراتب (من اليمين الى اليسار) وهي الاحاد ، العشرات ، المئات ، الالوف ، ... العدد في المرتبة الاولى (الاحاد) يكون

مضروباً في 10^0 ، والعدد في المرتبة الثانية (العشرات) مضروباً في 10^1 ، والعدد في مرتبة الثالثة (المئات) مضروباً في 10^2 . . . وهكذا . . . وان مجموع حواصل ضرب عشرات مختلفة الأس في المعاملات التي يتكون منها العدد تساوي العدد في النظام العشري .

مثال

العدد 125 بالنظام العشري يكتب بالشكل التالي

في المرتبة الأولى $5 = 5 * 10^0$ الاحاد

في المرتبة الثانية $2 = 2 * 10^1$ العشرات

في المرتبة الثالثة $1 = 1 * 10^2$ المئات

$$125_{(10)} = 1 * 10^2 + 2 * 10^1 + 5 * 10^0$$

مثال

العدد 571.23 يكتب بالنظام العشري (الموسع) بالشكل التالي :

الجزء الصحيح بالنظام العشري

$$571_{(10)} = 5 * 10^2 + 7 * 10^1 + 1 * 10^0$$

الجزء الكسري بالنظام العشري

$$.23 = 2 * 10^{-1} + 3 * 10^{-2}$$

$$571.23 = 5 * 10^2 + 7 * 10^1 + 1 * 10^0 + 2 * 10^{-1} + 3 * 10^{-2}$$

مثال

العدد (007) . بالنظام الموسع يكتب بالشكل التالي :

$$.007_{(10)} = 0 * 10^{-1} + 0 * 10^{-2} + 7 * 10^{-3}$$

2 . 3 . 1 النظام الثنائي BINARY SYSTEM

الرموز المستخدمة في هذا النظام هي الاعداد 0 و 1 والاساس هو 2 . يعتبر هذا النظام من اكثر الانظمة المستخدمة في تصميم الحاسبات الالكترونية في الوقت الحاضر ، اذ ان الحاسبات الالكترونية تعمل بالكهرباء وتعلم ان الدائرة الكهربائية تكون في احدى الحالتين ، فهي اما متصلة (ON) او منفصلة (OFF) ويرمز ب 0 اذا كانت الدائرة منفصلة و 1 اذا كانت متصلة .

مثال

الاعداد التالية هي بالنظام الثنائي

$$10001_{(2)} \quad 111100_{(2)} \quad 10101_{(2)}$$

فيما يلي استعراض للعمليات الجبرية في النظام الثنائي :

1- الجمع ADDITION

لجمع اي عددين في النظام الثنائي تتبع القواعد التالية :

+	0	1
0	0	1
1	1	10

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 10$$

مثال :

$$11011_{(2)} + 10010_{(2)} \text{ بالنظام الثنائي}$$

لايجاد

$$11011_{(2)} + 10010_{(2)} = 101101_{(2)}$$

$$\begin{array}{r}
 1 \\
 11011 \\
 + 10010 \\
 \hline
 101101
 \end{array}$$

تبدأ عملية الجمع (من اليمين الى اليسار) حيث يجمع كل رقم من العدد الاول مع نظيره من العدد الثاني ويكتب الناتج اسفل خط الجمع ، وفي حالة $1+1$ ، والناتج هو 10 يكتب 0 في نفس العمود ويضاف 1 الى العمود الذي يليه .

2- الطرح SUBTRACTION

لنطرح اي عددين في النظام الثنائي تتبع مايلي :

-	0	1
0	0	غير ممكن
1	1	0

$$\begin{array}{ll}
 0 - 0 = 0 & 1 - 0 = 1 \\
 0 - 1 = \text{غير ممكن} & 1 - 1 = 0
 \end{array}$$

مثال :

: لايجاد ناتج مايلي :

$$1- \quad 1100101_{(2)} - 1000011_{(2)}$$

$$\begin{array}{r}
 1 \\
 1100101 \\
 - 1000011 \\
 \hline
 0100010
 \end{array}$$

$$2- \quad 1000 \quad - \quad 111 \\ (2) \quad (2)$$

$$\begin{array}{r} 1 \\ 1000 \\ - 111 \\ \hline 0001 \end{array}$$

تبدأ عملية الطرح (من اليمين إلى اليسار) حيث يطرح كل رقم من العمود الثاني من نظيره من العمود الأول ويكتب الناتج أسفل خط الطرح ، وفي حالة طرح 0 من 1 وهو غير ممكن يأخذ 1 من العمود الذي يليه ويضاف إلى 0 فيصبح 10 ثم يطرح من 1 ويكون الناتج 1 كما في المثال أعلاه .

MULTIPLICATION

3- الضرب

لضرب أي عددين في النظام الثنائي تتبع مايلي :

*	0	1
0	0	0
1	0	1

$$0 * 0 = 0$$

$$1 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 1 = 1$$

مثال

لايجاد ناتج مايلي :

$$1011 \quad * \quad 101 \\ (2) \quad (2)$$

$$1011 \quad * \quad 101 \quad = \quad 110111 \\ (2) \quad (2) \quad (2)$$

$$\begin{array}{r} 1011 \\ * 101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 \end{array}$$

DIVISION القسمة 4

عند قسمة اي عددين في النظام الثنائي تتبع ماييلي :

$$0 \div 1 = 0$$

$$1 \div 1 = 1$$

مثال

لايجاد ناتج ماييلي :

$$1- \quad 11011_{(2)} \div 11_{(2)} = 1001_{(2)}$$

$$\begin{array}{r} 11 \overline{) 11011} \\ \underline{11} \\ 0001 \\ \underline{0001} \\ 0000 \\ \underline{0000} \\ 00000 \end{array}$$

$$2- \quad 1001_{(2)} \div 10_{(2)} = 10.1_{(2)}$$

$$\begin{array}{r} 11 \overline{) 1001} \\ \underline{10} \\ 0010 \\ \underline{0010} \\ 0000 \\ \underline{0000} \\ 00000 \end{array}$$

$$3- \quad 11101101_{(2)} \div 1001_{(2)} = 11010.01\dots_{(2)}$$

$$\begin{array}{r}
 \begin{array}{r}
 1001 \overline{) 11101101} \\
 \underline{1001} \\
 01011 \\
 \underline{1001} \\
 0001010 \\
 \underline{1001} \\
 0001100 \\
 \underline{1001} \\
 0000011
 \end{array}
 \end{array}$$

كما واضح في الامثلة اعلاه ان عمليتي الضرب والقسمة في الاعداد الثنائية مشابه تماما لعمليتي الضرب والقسمة في الاعداد العشرية

ملاحظة

- في النظام الثنائي العمليات الجبرية مشابه الى العمليات الجبرية في النظام العشري مع ملاحظة مايلي :
- 1 - عند جمع $1 + 1$ هو 10 (اي واحد صفر) وليس العدد عشرة اي اننا نكتب 0 ويكون 1 باليد ويضاف الى المرتبة التالية .
 - 2 - عند طرح 1 من 0 اي (0 - 1) فاننا نطلب 1 من المرتبة التي تلي 0 فتصبح القيمة (10) .
 - 3 - ان القسمة على الصفر غير ممكن في النظام الثنائي كما هي الحال في النظام العشري .

التحويل من نظام عددي الى نظام عددي اخر :

غالبا يكون من الضروري استخدام انظمة عددية مختلفة في مراحل مختلفة عند تنفيذ البرنامج ، لذا من الضروري معرفة كيفية تحويل اي عدد من نظام ما الى نظيره في نظام اخر .

فيما يلي موجز للتحويلات من نظام عددي الى اخر : -

التحويل من النظام الثنائي الى العشري :

ان تحويل اي عدد من النظام الثنائي الى نظيره في النظام العشري يتم عن طريق كتابة العدد بالنظام الموسع ، وبما ان الاساس في النظام الثنائي 2 لذلك فان كتابة العدد بالشكل الموسع يتم على شكل مجموع حاصل ضرب 2 وباس مختلف في معاملات العدد ، فان هذا المجموع يمثل العدد في النظام العشري .

مثال

نظير العدد $1101_{(2)}$ بالنظام العشري : -

$$\begin{aligned} 1101_{(2)} &= 1 * 2^0 + 0 * 2^1 + 1 * 2^2 + 1 * 2^3 \\ &= 1 + 0 + 4 + 8 \\ &= 13_{(10)} \end{aligned}$$

مثال

نظير العدد 110.01 بالنظام العشري : -

$$\begin{aligned} 110.01 &= 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ (2) &= 0 + 2 + 4 + 0 + 1 / 4 \\ &= 6.25 \end{aligned}$$

التحويل من النظام العشري الى الثنائي :

١- الاعداد الصحيحة

لتحويل العدد X من النظام العشري الى الثنائي يستخدم طريقة القسمة المتكررة على الاساس 2 ومن ثم تسجيل الباقي وحسب الخطوات التالية :

- 1- اكتب العدد X في العمود الاول والثاني R في العمود الثاني وهو عمود الباقي.
2. اقسم العدد X على الاساس 2 ، واكتب ناتج القسمة اسفل X في العمود الاول والباقي R في العمود الثاني مقابل ناتج القسمة
- 3- اذا كان ناتج القسمة يساوي صفرا ، توقف عن اجراء القسمة واكتب عمود الباقي R من الاسفل الى الاعلى في سطر ومن اليسار الى اليمين وبالتسلسل .

4- اذا كان ناتج القسمة لا يساوي صفرا ، تابع اجراء قسمة العدد في العمود الاول على الاساس 2 وكرر نفس الخطوات السابقة .

مثال

لتحويل العدد $22_{(10)}$ الى النظام الثنائي

$(X=)$ 22	R
11	0
5	1
2	1
1	0
0	1

$$22_{(10)} = 10110_{(2)}$$

مثال

لتحويل العدد $150_{(10)}$ الى النظام الثنائي

150	R
75	0
37	1
18	1
9	0
4	1
2	0
1	0
0	1

$$150_{(10)} = 10010110_{(2)}$$

فيما يلي جدول الاعداد من 1 الى 10 المكتوبة بالنظام العشري وما يقابلها بالنظام الثنائي :

النظام العشري	0	1	2	3	4	5	6	7	8	9	10
النظام الثنائي	0	1	10	11	100	101	110	111	1000	1001	1010

ب - الكسور

1 - يتم تحويل الكسور من النظام العشري الى الثنائي عن طريق عمليات الضرب الاعتيادية للكسر في الاساس 2 ويكون العامل صفرا اذا كان حاصل الضرب كسرا ويكون العامل هو الجزء الصحيح من ناتج الضرب اذا كان حاصل الضرب ليس كسرا .

2 - اذا كان ناتج الضرب يساوي عددا صحيحا تتوقف عندئذ عملية الضرب وتكتب الناتج من الاعلى الى الاسفل ومن اليسار الى اليمين وبالتسلسل .

مثال

لايجاد نظير الكسر 750. بالنظام الثنائي

$$\begin{array}{rcll} .750 * 2 & = & 1.50 & 1 \\ .5 * 2 & = & 1.0 & 1 \\ .750 & = & .11 & \\ (10) & & & \end{array} \quad \begin{array}{c} \downarrow \\ \downarrow \end{array}$$

(من الاعلى الى الاسفل)

مثال

نظير الكسر 225. بالنظام الثنائي

$$\begin{array}{rcll} .225 * 2 & = & .450 & 0 \\ .450 * 2 & = & .900 & 0 \\ .9 * 2 & = & 1.8 & 1 \\ .8 * 2 & = & 1.6 & 1 \\ .9 * 2 & = & 1.8 & 1 \\ . & & & \\ . & & & \\ . & & & \end{array} \quad \begin{array}{c} \downarrow \\ \downarrow \\ \downarrow \end{array}$$

.225 = .00111...

مثال

لتحويل العدد 57.125 الى النظام الثنائي

57	R
28	1
14	0
7	0
3	1
1	1
0	1

$$\begin{array}{rcl}
 .125 * 2 & = & .25 \quad 0 \\
 .25 * 2 & = & .5 \quad 0 \\
 .5 * 2 & = & 1.0 \quad 1
 \end{array}$$

$$57 = 111001_{(2)}$$

$$.125 = .001_{(2)} \quad (10)$$

$$57.125_{(10)} = 111001.001_{(2)}$$

1.3.3 النظام الثماني OCTAL SYSTEM

ان عدد الرموز المستخدمة في هذا النظام ثمانية من 0 الى 7 اي (0, 1, 2, 3, 4, 5, 6, 7, 8) وان الاساس هو العدد 8.

امثلة

العدد $251_{(8)}$ هو بالنظام الثماني

اي ان العدد 251 بالنظام الثماني لايساوي العدد 251 بالنظام العشري .

التحويل من النظام الثماني الى النظام العشري

ان تحويل الاعداد من النظام الثماني الى النظام العشري يتم بنفس طريقة تحويل الاعداد من النظام الثنائي الى العشري وباستخدام الاساس 8 بدلا من 2 .

مثال

العدد $251_{(8)}$ بالنظام العشري

$$251_{(8)} = 1 * 8^0 + 5 * 8^1 + 2 * 8^2 = 169_{(10)}$$

مثال

العدد 172.35 بالنظام العشري

$$\begin{aligned} 35_{(8)} &= 2 * 8^0 + 7 * 8^1 + 1 * 8^2 + 3 * 8^{-1} + 5 * 8^{-2} \\ &= 2 + 54 + 64 + 3/8 + 5/64 \\ &= 120 + 3/8 + 5/64 \\ &= 120 + 29/64 \end{aligned}$$


التحويل من النظام العشري الى النظام الثماني

يمكن تحويل الاعداد من النظام العشري الى النظام الثماني بنفس طريقة تحويل الاعداد من النظام العشري الى الثنائي وبالقسمة المتكررة على الاساس 8 بدلا من 2 ، وتسجيل الباقي في عمود الباقي (R) .

مثال

لتحويل العدد 125 الى النظام الثماني

125	R
15	5
1	7
0	1



$$125_{(10)} = 175_{(8)}$$

مثال

لتحويل العدد 128.25 الى النظام الثماني

128	R
16	0
2	0
0	2

$$128_{(10)} = 200_{(8)}$$

$$\begin{aligned} .25 \times 8 &= 2.00 & 2 \\ .25 &= .2 & (8) \\ & & (10) \end{aligned}$$

$$128.25_{(10)} = 200.2_{(8)}$$

إذن

كما يمكن تحويل الاعداد من النظام الثماني الى النظام الثنائي وبالعكس .

الجدول التالي يبين الاعداد من 0 الى 7 وما يقابله بالنظام الثنائي :

النظام الثماني	0	1	2	3	4	5	6	7
النظام الثنائي	000	001	010	011	100	101	110	111

مثال

لتحويل العدد $25_{(8)}$ المكتوب في النظام الثماني الى ما يقابله في النظام الثنائي

2	5	التماسي
010	101	الحدائي

$$25_{(8)} = 010101_{(2)}$$

وإن $25_{(8)} = 21_{(10)}$ برهن ذلك ؟

مثال

لتحويل العدد 734 من النظام الثماني الى ما يقابله في النظام الثنائي

7	3	4	الثماني
111	011	100	الثنائي

$$734_{(8)} = 111011100_{(2)}$$

ملاحظة

- 1 - يجب كتابة ثلاثة رموز لكل عدد ثنائي في كل حقل (الرقم 3 هو العدد 011 وليس 11) ، ثم كتابة العدد المقابل من اليسار الى اليمين .
- 2 - ان طريقة تحويل العدد من الثنائي الى الثماني تتم بالعكس .

مثال

تحويل العدد 11100110 الى الثماني يكون بالشكل التالي .

11	100	110	الثنائي
3	4	6	الثماني

$$11100110_{(2)} = 346_{(8)}$$

اي يتم توزيع كل ثلاثة رموز في حقل واحد (من اليمين الى اليسار) .

1.3.4 النظام السادس عشر HEXAD ECIMAL SYSTEM

يتكون هذا النظام من ستة عشر رمزا وهي الاعداد من 0,1,2,...,9 والحروف A , B , C , D , E , F وان الاساس لهذا النظام هو العدد 16

فيما يلي الرموز في النظام السادس عشر وما يقابلها في النظام العشري :

النظام العشري	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
النظام السادس عشر	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

التحويل من النظام السادس عشر الى الانظمة الاخرى :

- 1- للتحويل من النظام السادس عشر الى النظام العشري نتبع قاعدة الضرب المتكررة .
وللتحويل من النظام العشري الى النظام السادس عشر نتبع قاعدة القسمة المتكررة
كما هي الحال في تحويلات الانظمة الاخرى .
- 2- للتحويل من النظام السادس عشر الى الثنائي او بالعكس يتم توزيع الرموز الى مراتب حيث ان كل رمز او عدد في النظام السادس عشر يقابل اربع مراتب في النظام الثنائي ، وحسب الجدول التالي :

السادس عشر	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
الثنائي	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

مثال

يكون تحويل العدد $2C_{(16)} . 25A_{(16)}$ الى النظام العشري كما يلي :

$$\begin{aligned}
 25A.2C_{(16)} &= A * 16^0 + 5 * 16^1 + 2 * 16^2 + 2 * 16^{-1} + C * 16^{-2} \\
 &= 10 * 16^0 + 5 * 16^1 + 2 * 16^2 + 2 * 16^{-1} + 12 * 16^{-2}
 \end{aligned}$$

مثال

$$\overleftarrow{10010011101} \cdot \overrightarrow{0010} = 495.2_{(16)} \quad (2)$$

حيث ان :

0010 يقابل العدد 2 في النظام السادس عشر
1101 يقابل الحرف D في النظام السادس عشر
1001 يقابل العدد 9 في النظام السادس عشر
0100 يقابل العدد 4 في النظام السادس عشر

ملاحظة

هناك برامج للتحويل بين الانظمة المختلفة ضمن التمارين العامة .

تمارين

1 - جد ناتج مما يلي بالنظام الثنائي :

- 1- $10001 + 1101$
- 2- $100011 - 11100$
- 3- $1110 * 110$
- 4- $11011 \div 101$

2 - حول العدد 341 من النظام العشري الى :

ا - النظام الثنائي

ب - النظام الثماني

ج - النظام السادس عشر

3 - حول العدد 21.175 من النظام الثماني الى النظام العشري .

4 - حول العدد 5AB.2C من النظام السادس عشر الى النظام الثنائي

5 - حول العدد 1001101101 من النظام الثنائي الى النظام الثماني ثم الى النظام السادس عشر .

الفصل الثاني

الخوارزميات والمخططات الانسيابية

1 . 2 الخوارزميات ALGORITHMS

لقد اشرنا سابقا الى ان الحاسبة الالكترونية تتميز بقدرتها الفائقة في انجاز العمليات الحسابية حسب الاوامر والتعليمات المعطاة لها وكذلك امكانياتها في حفظ المعلومات المختلفة والتي يعجز الانسان عن حفظها واستعادتها عند الحاجة ، لكنها لا تستطيع القيام بشكل ذاتي لحل اي مسألة مهما كانت بسيطة ، لذا من الضروري توضيح المسألة قبل البدء بكتابة البرنامج بأي لغة من لغات البرمجة وذلك بوضع خطوات متسلسلة ومفهومة وبسيطة لحل المسألة .

طرق توضيح المسألة

الخوارزمية

عبارة عن خطوات متسلسلة ومفهومة لحل المسألة المراد كتابة البرنامج لها او بمعنى اخر الخوارزمية عبارة عن التسلسل المنطقي لحل المسألة .

مثال 1

الخوارزمية لعملية جمع عددين (اي التسلسل المنطقي لجمع عددين) هي :

1 - لنفرض ان العدد الاول هو X

2 - لنفرض ان العدد الثاني هو Y

3 - لنفرض ان حاصل جمع العددين هو Z اي ان $Z=X+Y$

4 - اطبع قيمة Z (وهي القيمة المطلوبة)

5 - النهاية

هذا المثال البسيط يوضح انه لكي نحصل على مجموع أى عددين يجب اولا معرفة قيمة العددين ثم جمعهم وطبع الناتج .

مثال 2

الخوارزمية لطبع عدد ما ومربع ومكعب العدد.

- 1 - لنفرض ان العدد الاول هو A .
- 2 - لنفرض ان مربع العدد $(A * A)$ هو B ، اي ان $B = A * A$ ($*$ يمثل عملية الضرب).
- 3 - لنفرض ان مكعب العدد $(A * A * A)$ هو C اي ان $C = A * A * A$
- 4 - اطبع كل من A , B , C .
- 5 - النهاية.

مثال 3

الخوارزمية لعملية قسمة عددين هي

- 1 - لنفرض ان العدد الاول هو X .
- 2 - لنفرض ان العدد الثاني هو Y .
- 3 - اذا كانت قيمة العدد الثاني تساوي صفر ($Y=0$) اذهب الى الخطوة رقم (7).
- 4 - اذا كانت قيمة Y لا تساوي صفر ، اذهب الى الخطوة رقم (5).
- 5 - لنفرض ان حاصل قسمة X / Y هو Z اي ان $Z = X / Y$ (/ : يمثل عملية القسمة).
- 6 - اطبع قيمة Z .
- 7 - النهاية .

مثال 4

الخوارزمية لحل معادلة من الدرجة الثانية وإيجاد الجذور الحقيقية وبطريقة الدستور .

ملاحظة

معادلة من الدرجة الثانية معرفة كالآتي :

$$AX^2 + BX + C = 0$$

حيث ان A, B, C ثوابت وان الجذور بطريقة الدستور هي :

$$X_{1,2} = (-B \pm \sqrt{B^2 - 4AC}) / 2A$$

- 1 _ الثوابت هي A, B, C
- 2 _ معرفة قيم كل من A, B, C
- 3 _ تحديد قيمة R حيث ان $R = B^2 - 4AC$
- 4 _ اذا كانت قيمة R اقل من الصفر اذهب الى الخطوة رقم (11)
- 5 _ اذا كانت قيمة R يساوي صفر اذهب الى الخطوة رقم (8)
- 6 _ في هذه الحالة تكون قيمة R اكبر من الصفر تكون الجذور

كالآتي

$$X2 = (-B + R) / 2A$$

$$X1 = (-B - R) / 2A$$

- 7 _ اذهب الى الخطوة رقم (9)
- 8 _ في هذه الحالة تكون الجذور كالآتي :
- 9 _ اطبع قيمة $X1, X2$
- 01 _ اذهب الى الخطوة رقم (12)
- 11 _ اطبع (لا توجد جذور حقيقية للمعادلة) .
- 21 _ النهاية

FLOW CHART

2.2 المخططات الانسيابية

ان المخطط الانسيابي يمثل وصفا تصويريا لخطوات الخوارزمية وتكون اكثر وضوحا من الخوارزمية ويقوم مقامها ، ويمكن بواسطته ملاحظة تتبع التسلسل المنطقي لحل المسألة بكل سهولة .

FLOW CHART SYMBOLS

رموز المخطط الانسيابي

الجدول التالي يحتوي على الرموز الخاصة المستخدمة عند رسم المخطط الانسيابي:

الرموز SYMBOLS

الاستعمال USAGE



بداية او نهاية البرنامج



لادخال و اخراج المعلومات



عملية حسابية



اتجاه سير البرنامج



اتخاذ قرار



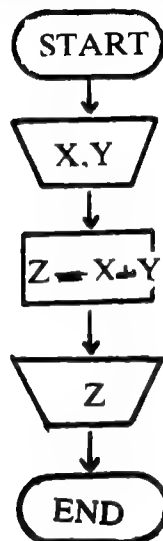
نقطة الانتقال او الاستمرار



تكرار او الدوران

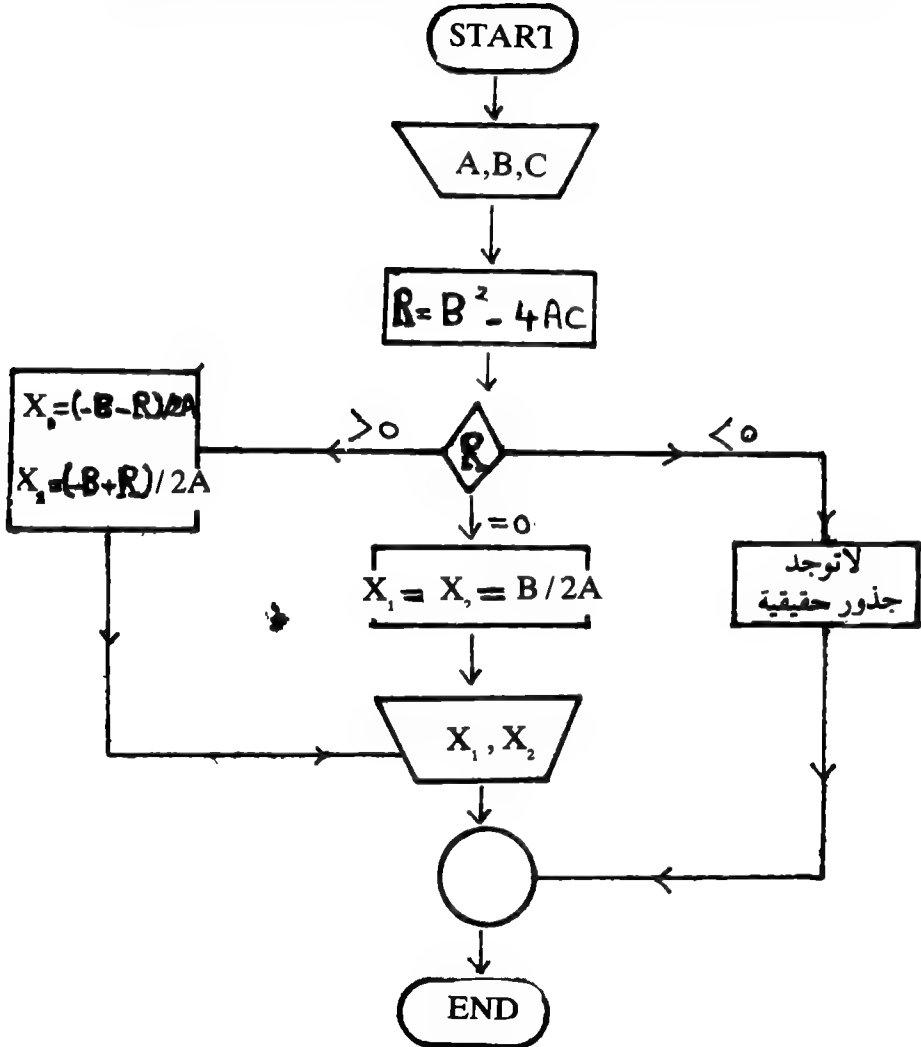
مثال 1

المخطط الانسيابي للمثال (1) اي لمجموع عددين .



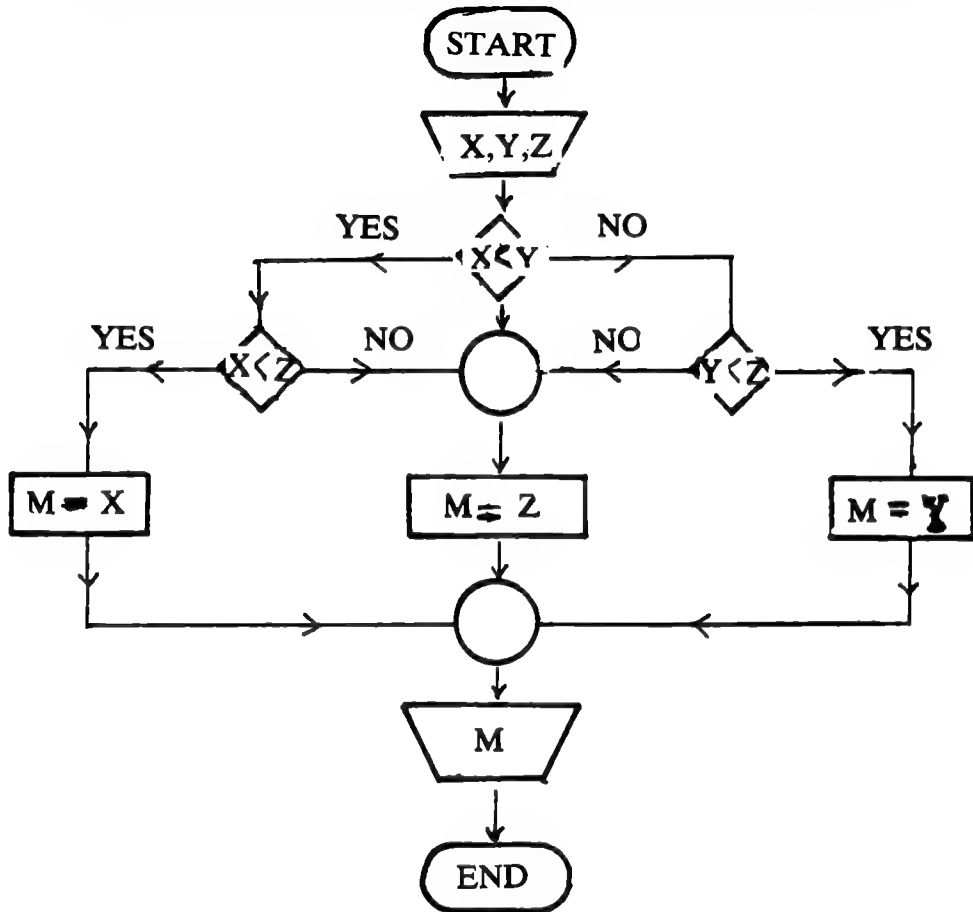
مسال 2

المخطط الانسيابي لايجاد الجذور الحقيقية في المثال (4) اي لمجموع عددين.



مثال 3

المخطط الآتيلي لايجاد اصغر عدد بين ثلاثة اعداد (راجع البرنامج في ص (٧٠))



تمارين

1 - اكتب الخوارزمية لإيجاد قيمة الدالة $F(X) = |X|$ حيث أن :

$$F(X) = \begin{cases} X & \text{IF } X \geq 0 \\ -X & \text{IF } X < 0 \end{cases}$$

- 2 - ارسم المخطط الانسيابي للسؤال الاول .
- 3 - ارسم المخطط الانسيابي لإيجاد أكبر عدد من بين مجموعة اعداد..
- 4 - اكتب الخوارزمية لطبع الاعداد الفردية ومكعباتها من 1 إلى 50 ثم ارسم المخطط الانسيابي للخوارزمية .
- 5 - ارسم المخطط الانسيابي لإيجاد الوسط الحسابي لعشرة اعداد .

الفصل الثالث

لغة البيسك

اساسيات لغة البيسك

3.2.2 الثوابت والمتغيرات

الثوابت CONSTANTS

الثابت عبارة عن مقدار يبقى ثابتا خلال فترة تنفيذ البرنامج وهناك نوعان من الثوابت:

١ - الثوابت العددية NUMERICAL CONSTANT

الثابت العددي عبارة عن عدد حقيقي قد يحتوي على الإشارة (+) اذا كان موجبا او (-) اذا كان سالبا .

امثلة 562 , +3.170 , -10.6

ملاحظة

لغة البيسك لا تسمح للعدد ان يتجاوز عدد ارقامه عن ٩ ارقام معنوية (SIGNIF).
عندما يكون العدد مقدار صغير جدا او كبير جدا يمكن استخدام طريقة
اخرى لكتابة العدد وهي طريقة التدوين البائي E-NOTATION . والصيغة
العامة لتمثيل العدد بهذه الطريقة هي :

$$\pm X E \pm n = \pm X * 10^{\pm n}$$

حيث ان X هو ثابت عددي ، و n عدد صحيح لا يتجاوز مرتبتين بالاضافة الى
مرتبة الإشارة ، تعتمد قيمة n على نوع الجهاز بصورة عامة تكون قيمة n
محصورة بين +38 و -38

مثال

العدد 1520000 يكتب بالشكل التالي

$$\begin{aligned} 1520000 &= 152 * 10^4 \\ &= +152E4 \end{aligned}$$

مثال

العدد 00471 . - يكتب بالشكل التالي :

$$\begin{aligned} -.00471 &= -471 * 10^{-5} \\ &= -471E-5 \\ &= -4.71 * 10^{-4} \\ &= -4.71E-4 \end{aligned}$$

ب - الثوابت الغير عددية (الثوابت الرمزية) NON - NUMERICAL
CONSTANT

عبارة عن مجموعة رموز محصورة بين علامتي الاقتباس ، وان طول الثابت الرمزي يعتمد على نوع الجهاز المستخدم .

امثلة "ABC", "12YX", "257" وقد يحتوي الثابت الرمزي على فراغات مثل
"AB 2D"

VARIABLES

المتغيرات

المتغير عبارة عن اسم لموقع من مواقع الذاكرة لحزن القيم . المتغير ممكن ان ياخذ اية قيمة تحدد له في البرنامج . والمتغيرات نوعان : -

١ - المتغيرات العددية NUMERICAL VARIABLES

تستخدم لحزن الثوابت العددية فقط ، والمتغير العددي يتكون من حرف ابجدي لواحد A - Z او حرف ابجدي واحد متبوعا برقم عشري واحد .
امثلة : X , X1 , A5 , B7

ب - المتغيرات الغير عددية NON NUMERICAL VARIABLES

تستخدم لحزن الثوابت الغير عددية ، والمتغير الغير عددي يتكون من حرف ابجدي واحد متبوع باشارة \$ او حرف مع رقم متبوع باشارة \$.
امثلة : X\$, X1\$, A5\$

ملاحظات :

١ - لا يجوز ان يتقدم الرقم عن الحرف في تسمية المتغيرات . مثلا 2X ليس اسما للمتغير .

٢ - ليس من الضروري ان يكون اسم المتغير متكونا من حرف واحد فقط ، حيث ان الاجهزة المتقدمة تسمح باكثر من حرف واحد ورقم واحد وحسب نوع الجهاز .

امثلة : SUM , AB2\$, AHMAD , AAAA3

3.2.3 العمليات والتعابير الرياضية ARITHMETIC EXPRESSIONS AND OPERATIONS

يتكون التعبير الحسابي من مجموعة من الثوابت العددية او المتغيرات العددية بالاضافة الى الاشارات الجبرية ، ومن العمليات الجبرية في لغة البيسك :

1- الجمع + $A + B$

2- الطرح - $A - B$

$A * B$	$*$	3-الضرب
A / B	$/$	4-القسمة
$A ^ B$	$^$	5-الرفع الى اس

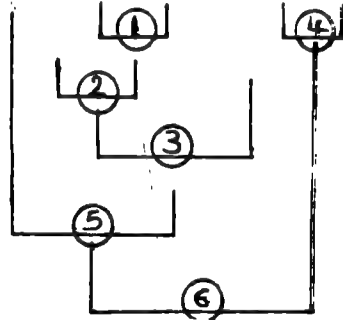
قاعدة الاسبقية PRECEDENCE

لتنفيذ التعبيرات الحسابية في لغة البيسك يجب اتباع قاعدة لتسلسل تنفيذ العمليات الحسابية تسمى قاعدة الاسبقية وتكون على النحو التالي :

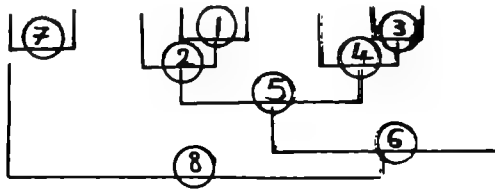
- 1- استخراج قيم التعبيرات المحصورة بين الاقواس اولا ومن اليسار الى اليمين .
- 2- ايجاد عمليات الرفع الى قوة ومن اليسار الى اليمين .
- 3- ايجاد عمليات الضرب او القسمة ومن اليسار الى اليمين .
- 4- ايجاد عمليات الجمع او الطرح ومن اليسار الى اليمين .

مثال : تكون اسبقية تنفيذ العمليات الحسابية التالية كالآتي :

1- $X - (2 + 5 * Y) ^ 2 + Z / 5$



2- $X - 2 * 5 / ((3 - 4 ^ 2) + (4 / (2 - 1) ^ 2))$



4.2.3 التعبير المنطقية LOGICAL EXPRESSION

يتكون التعبير المنطقي من مجموعة رموز وتستخدم لمقارنة اثنين من التعبيرات من نفس النوع ، ومن هذه الرموز :

$A=B$	$=$	- 1
$A<>B$	\neq	- 2
$A> B$	$>$	-3
$A < B$	$<$	-4
$A = > B$	\geq	-5
$A = < B$	\leq	-6

3.2.5 الايعارات

عبارة عن مجموعة اوامر او تعليمات تعطى للحاسبة من قبل المبرمج عن طريق وحدة الادخال .

ترقيم العبارات

كل عبارة مكتوبة بلغة البيسك تبدأ برقم يسمى رقم السطر ويكون تسلسل تنفيذ العبارات حسب ترقيم الاسطر حيث ان عبارة رقم 10 تتم تنفيذها قبل عبارة رقم 20. في البرنامج... وهكذا .

ملاحظة

عادة يترك فراغ واحد على الاقل بين كل عبارة ورقمها .

تمارين

1- اكتب التعبيرات الجبرية والعمليات الحسابية التالية بلغة البيسك :

$$1- \frac{5X^2Y}{X+Y} + \frac{6XYZ}{Z^3}$$

$$2- (XY - Z)^2 (2Z + X^3)^2$$

$$3- Y = (5X - 2)^2 / 3Z$$

2- بين تسلسل العمليات الحسابية للمعادلات التالية :

$$1- (2 * X * Z + 3 * Z^2) / (3 * Z + 1) + (X - Y)$$

$$2- 2 + 3 * X * (Z/K)^{(3/4)}$$

3- اكتب التعبيرات المنطقية التالية بلغة البيسك :

$$1- 2X + 3Y^2 \geq 0$$

$$2- (2X^2 - Y^2) \neq (X - Y)^3$$

الفصل الرابع

الاعزازات في لغة البيسك

الايعايات

اشرنا سابقا الى ان الايعايات هي عبارة عن مجموعة اوامر تعطى للحاسبة ، والايعايات في لغة البيسك نوعان :
أ - الايعايات التنفيذية : هي الايعايات التي تعطى للحاسبة لتنفيذ عمل معين .
ب - الايعايات غير التنفيذية : وهي الايعايات التي تزود المبرمج بمعلومات عن طبيعة البرنامج .
وفي هذه الايعايات :

4.1 ايعاز (اسناد) LET

هو ايعاز تنفيذي يستخدم لتحديد او اسناد قيمة تعبير حسابي معين الى متغير ما .
الصيغة العامة هي :

`N LET VARIABLE = ARITHMETIC EXPRESSION`

حيث ان N : عبارة عن رقم السطر

VARIABLE : اسم متغير

ARITH . EXP. : تعبير حسابي او ثابت عددي

```
10 LET X = 1
20 LET Y = X+5
```

مثال

حسب عبارة رقم 10 تقوم الحاسبة بتخصيص موقع من مواقع الذاكرة و اعطائه اسم X وتحديد قيمة لـ X يساوي واحد . اما عبارة 20 تعني ان الحاسبة تقوم بتخصيص موقع اخر في الذاكرة وتسميته باسم Y ومن ثم ايجاد قيمة التعبير الحسابي $X+1$ واعطاء القيمة الناتجة الى Y.

ملاحظة

يمكن الاستغناء عن ايعاز LET في العبارة بالنسبة للاجهزة المتقدمة وذلك لان المساواة تقوم مقام هذا اليعاز .

4.2 ايعاز (ادخل) INPUT

هو ايعاز تنفيذي يستخدم لادخال المعلومات الى داخل الجهاز ومن خلال وحدة الادخال .

الصيغة العامة هي :

N INPUT LIST (V1, V2, ...)

N : عبارة عن رقم السطر
LIST : هو قائمة باسماء المتغيرات المراد تعريفها و استخدام الفارزة للفصل بين هذه المتغيرات .

ملاحظة

عند تنفيذ ايعاز INPUT تظهر على الشاشة علامة استفهام (؟) تعني اعطاء المبرمج قيم للمتغيرات المعرفة بواسطة INPUT وذلك لتكملة تنفيذ البرنامج .

مثال

```
10 INPUT X
20 INPUT Y$
```

'ويمكن تعريف X , Y\$ بايعاز واحد هي

```
30 INPUT X و Y$
```

عند تنفيذ عبارة رقم 10 تظهر علامة استفهام واحدة تطلب قيمة عددية واحدة للمتغير X . عند تنفيذ عبارة رقم 20 تظهر علامة استفهام اخرى تطلب قيمة رمزية واحدة الى Y ، ولكن عند تنفيذ عبارة رقم 30 تظهر علامة استفهام تطلب قمتان بالتسلسل وتفصلهم فارزة ، الاولى قيمة عددية للمتغير X ، والثانية قيمة رمزية لـ Y\$.

ايعاز اقراء ~ البيانات READ ~ DATA

ايعاز READ ، ايعاز تنفيذي من ايعازات الادخال تستخدم لتعريف مجموعة متغيرات ويختلف عن ايعاز INPUT لكونه يحتاج الى ايعاز اخر وهو DATA لاعطاء القيم الى المتغيرات ضمن البرنامج (اى ان هذين الايعازين متلازمان في البرنامج) ، وان ' DATA ' هو ايجاز غير تنفيذي يحدد قيم المتغيرات ويمكن كتابته في اي موقع ضمن البرنامج . قبل ايعاز انتهاء البرنامج والصيغة العامة هي :

```
N   READ   LIST   ( VARIABLES )  
    ⋮  
M   DATA  LIST   ( CONSTANTS )
```

N : عبارة عن رقم السطر

LIST (VAR .) : اسماء المتغيرات المراد تعريفها

LIST (CON .) : قيم المتغيرات التي تعرف بواسطة READ

ملاحظة

يجب ان لا يقل عدد القيم المعطاة في ايعاز DATA عن عدد المتغيرات المشار اليها بواسطة READ.

مثال

```
10   READ   X  
20   READ   Y$  
    ⋮  
100  DATA  12  
110  DATA  'ALI'
```

يمكن كتابة نفس المثال بصيغة أخرى وهي :

```
10 READ X
20 READ Y$
   .
   .
50 DATA 12 , 'ALI'
```

او

```
10 READ X,Y$
   .
   .
50 DATA 12,"ALI"
```

ملاحظة

يفضل تعريف المتغيرات باستخدام READ اذاكثر عددها في البرنامج .

ايعاز (اطبع) PRINT

وهو ايعاز تنفيذي يستخدم لاستخراج قيم المتغيرات من وحدة الذاكرة وطبعها الى الوسط الخارجي من خلال وحدة الاخراج والصيغة العامة هي :

```
N PRINT LIST ( VARIABLES )
```

N : رقم العبارة

(. VAR) LIST : قائمة باسماء المتغيرات المراد استخراج قيمها والتي تفصلهم

(,) أو (;) .

ملاحظات

1 - يستخدم الرمز (;) للفصل بين المتغيرات ولكن المسافة تكون اكبر عند ..

استخدام الرمز (و) .

2 - لاستخراج وطبع قيم المتغيرات على الوسط الخارجي بعد تنفيذ البرنامج

يستخدم ايعاز LPRINT .

مثال

```
10 READ X و Y
20 LET Z=X+Y
30 PRINT Z
40 DATA 5 و 10
```

ملاحظة :

يمكن استخدام ايعاز PRINT لطبع قيمة عددية او عبارة رمزية غير مخزونة في الذاكرة .

مثال

```
10 PRINT 5
20 PRINT "HELLO"
```

ملاحظة

يمكن استخدام ايعاز PRINT لترك سطر في البرنامج .

مثال

اكتب برنامجا لقراءة متغيرين X , Y ثم اطبع قيمة X في السطر الاول وقيمة Y في السطر الثالث .

```
10 READ X و Y
20 PRINT X
30 PRINT
40 PRINT Y
50 DATA 10 و 20
60 END
```

4.6 ايعازات السيطرة (التحكم) CONTROL STATEMENTS

لقد اشرنا سابقا بان البرنامج عبارة عن مجموعة متسلسلة من الاوامر او الايعازات يتم تنفيذه حسب التسلسل الا ان هناك بعض المسائل يستوجب تغير سير

تنفيذ البرنامج من موقع الى اخر ويتم ذلك باستخدام ايعازات تسمى ايعازات السيطرة . اي ان ايعازات السيطرة هي الاوامر التي يتحكم في سير تنفيذ البرنامج . ومن ايعازات السيطرة : -

4.6.1 ايعاز الانتقال غير الشرطية (اذهب الى) GO - TO

ان استخدام هذا الایعاز يسبب انتقال غير شرطي من عبارة الى اخرى . و الصيغة العامة هي :

N GOTO L

N : رقم العبارة

L : رقم العبارة التي يجب الانتقال اليها تنفيذاً لعبارة GO TO

مثال

```

10 INPUT X, Y
20 PRINT X, Y
30 GOTO 100
40 LET Z=X+Y
:
:
100 END

```

في هذا المثال يتم تنفيذ البرنامج بالتسلسل ابتداء من عبارة رقم 10 الى رقم 30 . اما في عبارة رقم 30 فان ايعاز GOTO يسبب انتقالاً الى عبارة رقم 100 لتنفيذها بدلا من تنفيذ عبارة رقم 40 .

4.4 ايعاز ملاحظة REM

ان ايعاز REM من كلمة (REMARK) اي الملاحظة عبارة عن ايعاز غير تنفيذي يستخدم لتوضيح بعض فقرات البرنامج للمبرمج . الصيغة العامة هي :

N REM EXPRESSION

N : رقم السطر
EXP: الوصف المطلوب

ملاحظة

ان هذه العبارة تهمل من قبل الحاسبة .

4.5 ايعاز نهاية END

ايعاز غير تنفيذي يستخدم لانهاء البرنامج ويأتي في نهاية البرنامج عادة . الصيغة العامة هي :

N END

ملاحظة

ان كل ايعاز يمثل عبارة واحدة (سطر واحد) في لغة البيسك ، اي ان كل عبارة تحمل رقما واحدا تحتوي على ايعاز واحد فقط ، ويمكن ان يحتوي السطر الواحد على عدة عبارات يفصل بين كل عبارة وعبارة رمز (:) COLON ، علما بان السطر الواحد يمكن ان يطبع فيه خمسة وسبعون رمزا .

مثال

```
10 LET X=1
20 LET Y=X*2+5
30 PRINT X و Y
40 X =X+1 : GOTO 10
```

عند تنفيذ هذا البرنامج يتم ايجاد قيمة Y ثم طبع قيم X , Y وزيادة قيمة X بواحد ثم العودة الى بداية البرنامج بواسطة GOTO .

ملاحظة

(لاتوجد نهاية لهذا البرنامج) .

4.6.2 إيعاز اذا كان IF - THEN

يستخدم هذا الإيعاز لاختبار صحة شرط معين ثم اختيار السطر المطلوب تنفيذه اذا كان الشرط صحيحا . الصيغة العامة هي :

N IF CONDITION THEN M

IF : لاختبار صحة الشرط
COND. : الشرط المطلوب اختباره ويكون عادة تعبير منطقي
THEN: هو الجزء المكمل للعبارة الشرطية

M : عندما يكون الشرط صحيحا فان M يمكن ان يكون :

1 - رقم السطر المطلوب الانتقال اليه لتنفيذه

2 - عبارة من عبارات البيسك المطلوب تنفيذه

اما اذا كان الشرط غير صحيحا يتم تنفيذ العبارة التي تلي عبارة N مباشرة .

مثال

```
20 IF X=0 THEN 120
30 IF X>0 THEN Y=X
```

في عبارة رقم 20 , 120 يمثل رقم السطر المطلوب تنفيذه اذا كان الشرط صحيحا ،
اما اذا كان الشرط غير صحيحا يتم تنفيذ عبارة رقم 30 .

مثال

اكتب برنامجا لقراءة X , Y ثم ايجاد قيمة Z حيث :

$$Z = \begin{cases} X-Y & X > Y \\ 2 & X = Y \\ X+Y & X < Y \end{cases}$$

```

10  REM FIND THE VALUE OF Z
20  READ X, Y
30  IF X>Y THEN 50
40  Z= X+Y : GOTO 60
50  Z=(X-Y)/2
60  PRINT Z
70  END

```

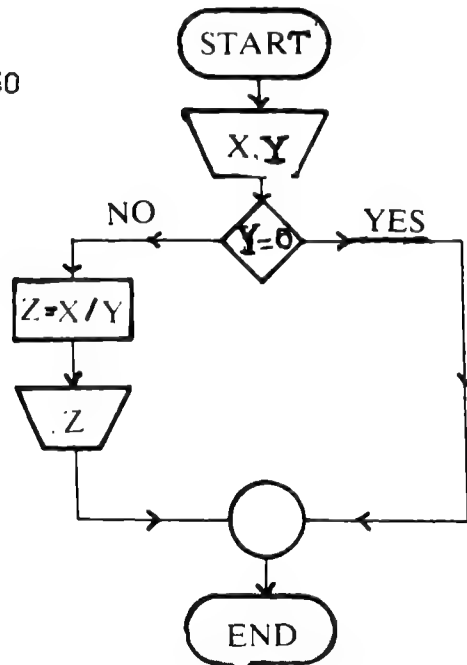
مثال

اكتب برنامجا لقراءة متغيرين وإيجاد حاصل القسمة بشرط ان المقام لا يساوي صفرا
ارسم المخطط الانسيابي للبرنامج .

```

10  INPUT X, Y
20  IF Y=0 THEN 50
30  Z=X/Y
40  PRINT Z
50  END

```



مثال

اكتب برنامجا بلغة البيسك لايجاد الجذور الحقيقية لمعادلة من الدرجة الثانية :
 $AX^2 + BX + C = 0$ واطبع (الجذور خيالية) عندما يكون الجذور غير حقيقية .

$$X_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad \text{حيث ان :}$$

```
10  REM  FIND THE REAL ROOTS FOR  AX2+BX+C
20  INPUT  A,B,C
30  R=B^2-4*A*C
40  IF  R<0  THEN  130
50  IF  R=0  THEN  90
60  X1=(-B+R^.5)/2*A
70  X2=(-B-R^.5)/2*A
80  GOTO  110
90  X1=-B/2*A
100  X2=X1
110  PRINT  X1,X2
120  GOTO  140
130  PRINT  'THE ROOTS ARE IMAJINARY'
140  END
```

4.6.3 ايعاز عند اذهب ON - GOTO

الصيغة العامة هي :

N ON M GOTO N1, N2, ...

... و N_1, N_2 : ارقام الاسطر يتم الانتقال الى واحدة منها اعتمادا على قيمة M. M: متغير عددي او تعبير حسابي يأخذ قيم متسلسلة ابتداء من واحد وتنفيذ العبارة يكون بالشكل التالي :
اذا كان قيمة M يساوي واحد يتم تنفيذ عبارة رقم N_1

إذا كان قيمة M يساوي اثنان يتم تنفيذ عبارة رقم M_2 ، وهكذا

مثال

اكتب برنامجا لاجاد قيمة y حسب المعادلة :

$$Y = \begin{cases} X^2 + 5X & \text{IF } X=1 \\ 0 & \text{IF } X=2 \\ X+2 & \text{IF } X=3 \end{cases}$$

```
10 INPUT X
20 ON X GOTO 30 , 40 , 50
30 Y=X^2+5*X : GOTO 60
40 Y=0 : GOTO 60
50 Y=X+2
60 PRINT Y
70 END
```

مثال

إذا كان لدينا مجموعة ارقام ، اكتب برنامجا لمعرفة الفردية منها والزوجية حسب

المعادلة $Y = X - 2 * \text{INT}(X/2) + 1$ ، يكون العدد X زوجيا إذا كان $Y = 1$

وفرديا إذا كان $Y = 2$.

```
10 INPUT X
20 LET Y=X-INT(X/2)*2+1
30 ON Y GOTO 40 , 60
40 PRINT X و "EVEN"
50 GOTO 10
60 PRINT X و "ODD" : GOTO 10
```

ملاحظة

INT عبارة عن دالة مكتتبية مأخوذة من كلمة INTEGER حيث انها تأخذ الجزء الصحيح من العدد الحقيقي ولمعرفة المزيد عن هذه الدالة راجع الصفحة (ص

(١٠٢)

ملاحظة

يمكن كتابة البرنامج السابق بطريقة اخرى باستخدام ايعاز IF-THEN بدلا من ON-GOTO وكالاتي :

```
10 INPUT X
20 LET Y=X-INT(X/2)*2
30 IF Y=0 THEN PRINT X ؛ "EVEN" : GOTO 10
40 PRINT X ؛ "ODD" : GOTO 10
```

مثال

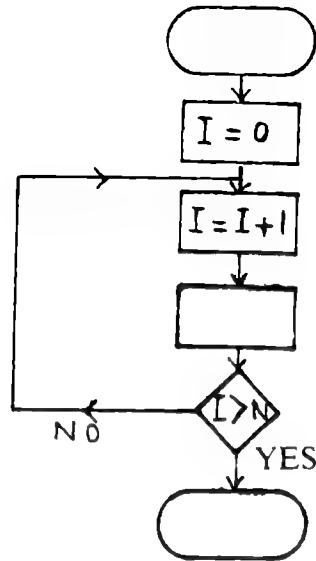
إذا كان X, Y, Z تمثل ثلاثة اعداد ، اكتب برنامجا لايجاد اصغر عدد بين الاعداد (لاحظ المخطط الانسيابي للبرنامج في ص ٤٥).

```
10 INPUT X,Y,Z
20 IF X<Y THEN 50
30 IF Y<Z THEN M=Y : GOTO 60
40 M=Z : GOTO 60
50 IF X<Z THEN M=X
60 PRINT " THE SMALLEST VALUE = " ؛ M
70 END
```

العداد COUNTER

عبارة عن متغير يدخل في البرنامج ليعد عدد التكرارات لمجموعة من الخطوات تسمى خوارزمية العد COUNTING ALGORITHM . والخطوات هي :

- 1 - افرض ان قيمة المتغير الذي يمثل العدد مساويا للصفر ($I=0$)
- 2 - اجعل القيمة الجديدة للعداد تساوي القيمة القديمة زائدا واحد اي ان ($I=I+1$)
- 3 - كرر الخطوات ابتداء من (2) الى ان تنتهي عملية العد وذلك بمقارنة قيمة I مع العدد النهائي للتكرار .



مثال

اكتب برنامجا لطبع الاعداد (0 - 100)

```

10 I=0
20 PRINT I
30 I=I+1
40 IF I<=100 THEN 20
50 END
  
```

ملاحظة

في هذا المثال I يمثل العداد وفي نفس الوقت يمثل العدد المطلوب طبعه .

مثال

اكتب برنامجا بلغة البيسك لقراءة خمس قيم لـ X وايجاد قيم Y حسب المعادلة :

$$Y = X^2 + 2X$$

```

10 LET I=1
20 INPUT X
30 Y=X^2+2*X
40 PRINT 'X=';X و 'Y=';Y
50 I=I+1
60 IF I<=5 THEN 20
70 END
  
```

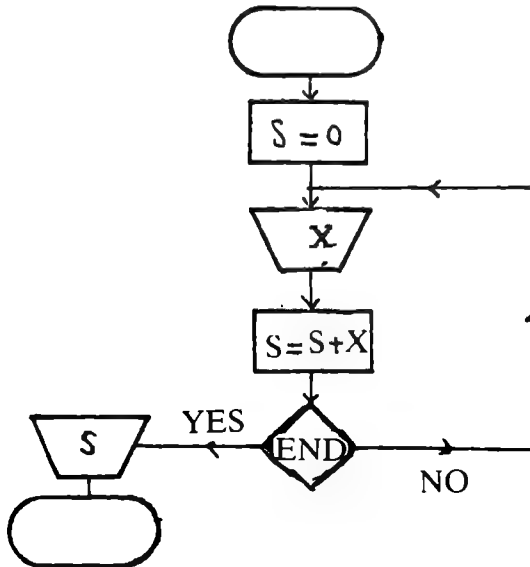

ملاحظة

- 1 - يمكن ايضا اختيار القيمة البدائية لـ I مساويا للصفر ($I = 0$) وتبعاً لذلك عبارة رقم 60 تتغير الى : 20 IF $I \leq 4$ THEN 60.
- 2 - في هذا المثال المتغير I يقوم مقام العداد فقط .

خوارزمية التجميع SUMMATION ALGORITHM

هي خطوات ايجاد مجموع مجموعة كبيرة من المتغيرات التي تمثل ظاهرة معينة باستخدام متغيرين اثنين فقط ، احدهما هو المتغير المراد جمعه والاخر هو المجموع وباستخدام ايعازات السيطرة و العداد .

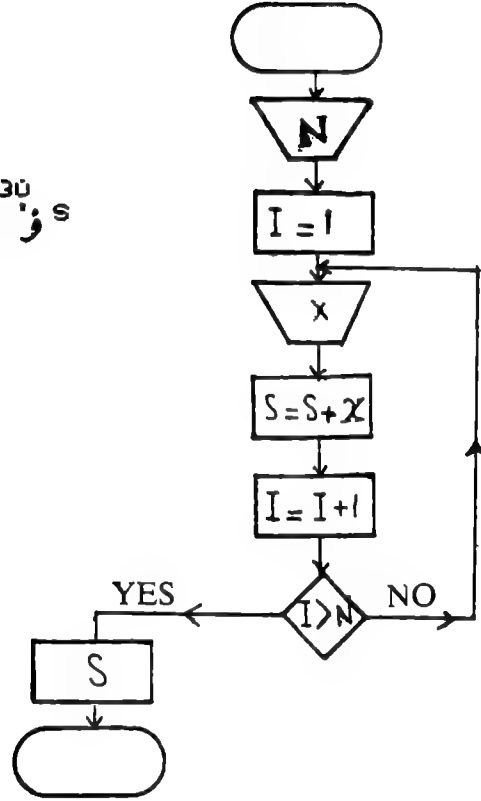
- 1 - افرض القيمة البدائية للمجموع يساوي صفراً ($S=0$)
 - 2 - اقرأ قيمة واحدة للمتغير X .
 - 3 - اجعل القيمة الجديدة لـ S تساوي القيمة القديمة له زائدا القيمة المعطى لـ X اي ان ($S = S + X$)
- كرر الخطوات ابتداءاً من خطوة رقم (2) الى انتهاء عملية قراءة القيم ، وتكون المخطط الانسيابي لهذه الخوارزمية كالآتي :



مثال

اكتب برنامجا لاجاد مجموع N من القيم وارسم المخطط الانسيابي لذلك .

```
10 INPUT N
20 I=1 : S=0
30 INPUT X
40 S=S+X
50 I=I+1
60 IF I<=N THEN 30
70 PRINT 'THE SUM= ' S
80 END
```



في العبارة رقم 20 المتغير I يمثل العداد الذي يعد عدد التكرارات في البرنامج و S يمثل المجموع . يمكن كتابة مجموعة اسطر على شكل سطر واحد باستخدام الرمز (:) COLON للفصل بين العبارات كما هو الحال في عبارة رقم 20 .

ملاحظة

عند ضرب مجموعة كبيرة من المتغيرات يمكن اتباع نفس قاعدة الجمع بجعل القيمة البدائية لحاصل الضرب مساوية الى واحد بدلا من الصفر (في حالة الجمع) ،
والعملية الحسابية هي عملية الضرب بدلا من الجمع .

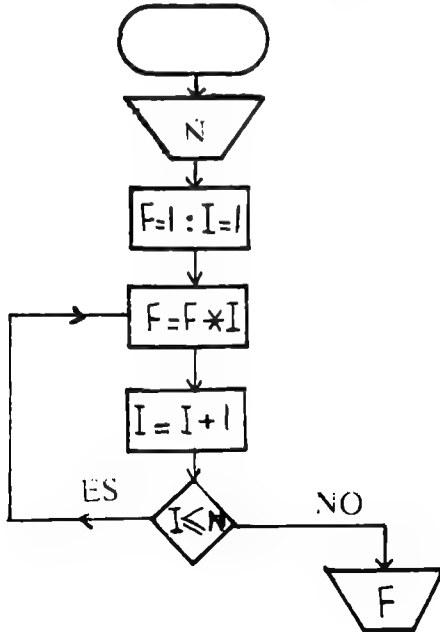
مثال

اكتب برنامجا لاجاد $N!$

حيث ان N عدداً صحيحاً $N! = N * (N-1) * (N-2) \dots 3 * 2 * 1$

```
10 INPUT N
20 F=1 : I=1
30 F=F*I
40 I=I+1
50 IF I<=N THEN 30
60 PRINT F : END
```

وان المخطط الانسيابي هو كما يلي :



4 . 6 . 4 ايعاز من الـ FOR ~ NEXT

هناك طريقة اخرى لتكرار تنفيذ مجموعة من الاوامر يحددها احدى اوامر السيطرة بدلا من الايعاز (IF ~ THEN) ولا تخرج من تكرار الاوامر لحين اكمال العدد او تحقيق الحالة المطلوبة والايعاز هو FOR-NEXT .

تكون الصيغة العامة كالآتي :

حلقة	{	N	FOR VARIABLE = M1 TO M2 STEP M3
		⋮	الاورام المراد تكرارها
		M	NEXT VARIABLE

VARIABLE : يمثل العدد الذي يعد عدد التكرارات ويتغير قيمته كلما تم تكرار الدورة .

M1 : القيمة البدائية للعدد .

M2 : القيمة النهائية للعدد .

M3 : مقدار الزيادة او النقصان في قيمة العدد خلال كل دوره .

ملاحظة

القيمة البدائية والنهائية والزيادة في ايعاز FOR -NEXT ممكن ان يكون ثوابت عددية او متغيرات عددية او تعبيرات حسابية .

أمثلة

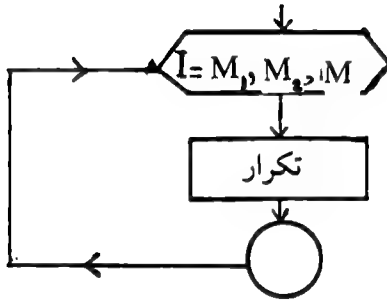
FOR I = M+2 TO M*N

FOR R = 0.05 TO 0.25 STEP .01

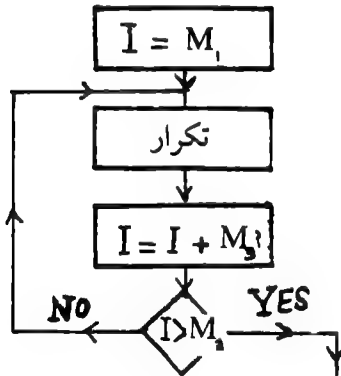
FOR K = 20 TO 5 STEP -1

الحلقة (LOOP)

مجموعة الاوامر المراد تكرارها باستخدام ايعاز FOR - NEXT يمكن توضيح صيغة FOR - NEXT كالآتي :



اما باستخدام IF - THEN هي :



لاحظ ان عدد الخطوات باستخدام FOR - NEXT يقل عن عدد الخطوات باستخدام IF - THEN .

يمكن توضيح الحلقات الصحيحة الممكن استخدامها بالبرمجة والحلقات الخاطئة الغير المسموح بها كالآتي :

حلقات صحيحة

```

[ FOR I
  [ FOR J
    [ FOR K
      NEXT K
    FOR K
      NEXT K
    FOR J
      NEXT J
  FOR J
    NEXT J
  FOR I
    NEXT I
  ]
]
  
```

حلقات خاطئة (متشابكة)

```

[ FOR I
  [ FOR J
    [ FOR K
      NEXT I
    FOR K
      NEXT J
    FOR J
      NEXT I
  ]
]
  
```

ملاحظة

إذا علم عدد التكرارات لاية عملية في البرنامج يستخدم ايعاز (FOR - NEXT) اي عند معرفة القيمة البدائية والنهائية للعداد ، اما اذا كانت القيمة النهائية للعداد غير معلومة فيستخدم ايعاز GOTO .

4.6.5 امثلة

مثال 1

اكتب برنامجا لقراءة الاعداد 102 , 11 , 37 , 12 , 5 ثم اطبع الاعداد الاولى منها .

(العدد الاول هو العدد الذي لا يقبل القسمة الا على نفسه وعلى واحد)

```

10 REM PRINT THE PRIME NUMBER
20 FOR I=1 TO 5
30 READ X
40 FOR J=2 TO X-1
50 R = X-INT(X/J)*J
60 IF R=0 THEN 90
70 NEXT J
80 PRINT X
90 NEXT I
100 DATA 5, 12, 37, 11, 102 : END
  
```

مثال 2

اكتب برنامجا لاجاد مساحة المثلثات القائمة الزوايا التي تبلغ قواعدها 2 , 10 , 7 , 14 , 12 وارتفاعاتها 4 , 20 , 15 , 8 , 3 على التوالي

1 - باستخدام الحلقات

2 - دون استخدام الحلقات

باستخدام الحلقات

```
10 FOR I=1 TO 5
20 PRINT 'INPUT THE BASE AND THE HIGHT '
30 READ X,Y
40 LET A=.5*X*Y
50 PRINT 'THE AREA = 'A
60 NEXT I
70 DATA 2,4,10,20,7,15,14,8,12,3
80 END
```

دون استخدام الحلقات :

```
10 I=1
20 PRINT 'INPUT THE BASE AND THE HIGHT'
30 INPUT X,Y
40 LET A=.5*X*Y
50 PRINT ' THE AREA = 'A
60 I=I+1
70 IF I<=5 THEN 20
80 END
```


مثال 3

اكتب برنامجا لقراءة المتغير X ثم ايجاد المجموع التالي :

$$S = X - X^3/3 - X^5/5 - X^7/7 + \dots - X^{15}/15$$

```
10 S=0 : K=0
20 INPUT X
30 FOR I=1 TO 15 STEP 2
40 S =S+(-1)^K*X^I/I
50 K =K+1
60 NEXT I
70 PRINT 'THE SUM = ' ;S
80 END
```

مثال 4

اكتب برنامجا لطبع الحرفين M , N بحيث ان يكون بينهم ثلاثة أسطر .

```
10 PRINT "M"
20 FOR I=1 TO 3
30 PRINT :NEXT I
40 PRINT "N"
50 END
```

مثال 5

اكتب برنامجا لقراءة المتغير X ثم جد قيمة EXP (X) حيث ان :

$$EXP(X) = 1 + X/1! + X^2/2! + \dots + X^N/N!$$

```
10 S=1
20 INPUT X, N
30 FOR I=1 TO N
40 F =1
50 FOR J =1 TO I
60 F =F*J : NEXT J
70 S =S+X^I/F
80 NEXT I
90 PRINT 'S= ' ;S
100 END
```

...

ملاحظة

ان اي عبارة من العبارات الاساسية في لغة البيسك تبدأ باحدى الكلمات التالية :

INPUT(READ), LET, PRINT, IF, GOTO, FOR, END

تمارين

1 - اكتب برنامجا لقراءة قيم X, Y واطبع قيمة Z حيث :

$$Z = Y \sum_{i=1}^{10} X^i$$

2 - اكتب برنامجا لقراءة عددين X, Y ثم عين فيما اذا كان كل عدد زوجي د فردي .

3 - اكتب برنامج لقراءة قيمة a ثم احسب قيمة b واطبع a, b حيث :

$$b = \begin{cases} a+2 & \text{IF } a < 0 \\ \frac{a^2 + 2a}{3} & \text{IF } 0 < a < 1 \\ \sqrt{a+3} & \text{IF } a > 1 \end{cases}$$

4 - اكتب برنامجا يقوم بالعمليات التالية :

- 1 - طبع الاعداد الفردية التي تقبل القسمة على 3 والتي تقع بين 10 - 100 .
- 2 - جمع الاعداد الصحيحة والتي تقع بين 10 - 100 .
- 3 - حاصل ضرب الاعداد الزوجية والتي تقع بين 10 - 100

الفصل الخامس

المصرفيات

المصفوفات

لا بد من الإشارة الى ان المتغيرات التي سبقت ذكرها كانت من نوع المتغير البسيطة . ويقصد بالمتغير البسيط ذلك المتغير الذي يحجز موقع واحد. في ذاكرة الحاسبة الالكترونية ، ولكن عندما تكون المتغير عبارة عن مجموعة عناصر من نفس النوع تخزن في مجموعة فلايا في الذاكرة وتحت اسم واحد يسمى بالمتغير المؤشر او المصفوفة .

المصفوفة

المصفوفة عبارة عن كميات او اعداد مرتبة في صفوف و اعمدة على شكل مستطيل وتكتب بالشكل التالي

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & a_{M3} & \dots & a_{MN} \end{pmatrix}_{M \times N}$$

كل من الكميات او الاعداد $a_{11}, a_{12}, \dots, a_{MN}$ والتي تشكل هذه المجموعة يسمى عنصرا في المصفوفة ويطلق على الارقام المرافقة لعناصر المجموعة الادلة او المؤشر الذي يعين موقع العنصر في المصفوفة

1 . 5 مصفوفة ذات بعد واحد

ONE-DIMENSIONAL ARRAY

عبارة عن مجموعة عناصر مرتبة على شكل صف واحد او عمود واحد ويوجد مؤشر واحد يشير الى تسلسل العنصر في المصفوفة .

مثال

$$X = X_1, X_2, \dots, X_N$$

المصفوفة X تتكون من مجموعة عناصر X_1, X_2, \dots, X_N

مرتبة على شكل صف واحد وان :

X_1 : هو العنصر الاول في المصفوفة

X_5 : هو العنصر الخامس في المصفوفة

X_N : هو العنصر الاخير في المصفوفة

وصيغة الكتابة بلغة البيسك هي كالآتي : $X(1), X(2), X(3), \dots$

او $X(I)$ حيث I يأخذ القيم من 1 الى N ، I يسمى مؤشر المصفوفة ويشير الى موقع كل عنصر في المصفوفة .

يتم حجز الخلايا المطلوبة لحزن عناصر المصفوفة في الذاكرة الرئيسية باستخدام ابعاد DIM من $DIMENSION$ وهو ايعاز غير تنفيذي والصيغة العامة هي :

... و NAME2(M) , NAME1(N) DIM N

حيث NAME1 : هي مصفوفة عدد عناصرها N

NAME2 : هي مصفوفة عدد عناصرها M .

N, M : ابعاد المصفوفات

مثال

10 DIM X(20) , A\$(30)

ملاحظة يجب تحديد عدد الخلايا المطلوبة قبل قراءة عناصر المصفوفة .

قراءة المصفوفة ذات بعد واحد

إذا كان لدينا مصفوفة $A(N)$ ثم قراءة وتخزين عناصر المصفوفة في الذاكرة باستخدام الحلقات (بعد تحديد الخلايا لكل عنصر) وبالشكل التالي :

```
FOR I=1 TO N
INPUT (READ) A(N)
NEXT I
```

كتابة المصفوفة ذات بعد واحد

يتم كتابة المصفوفة $A(N)$ المخزونة في الذاكرة بنفس طريقة قراءة وبالشكل التالي :

```
FOR I=1 TO N
PRINT A(I)
NEXT I
```

مثال

اكتب برنامجاً لقراءة القيم 3، 8، 7، 21، 10 المرتبة على شكل مصفوفة أحادية ثم اطبع عناصر المصفوفة .

```
10 REM READ AND PRINT AN ARRAY
20 DIM A(5)
30 FOR I=1 TO 5
40 READ A(I) : NEXT I
50 FOR I=1 TO 5
```

```
60 PRINT A(I) : NEXT I
```

```
70 DATA 10, 12, 7, 8, -3
```

```
80 END
```

ملاحظة

عدد الخلايا المطلوب حجزها باستخدام DIM يجب ان يكون مساويا على الاقل لعدد العناصر في المصفوفة .

2. مصفوفة ذات بعدين (ثنائية)

TWO DIMENSIONAL ARRAY

عبارة عن مجموعة عناصر مرتبة على شكل مجموعة من الصفوف ومجموعة من الأعمدة . مؤشرات الأول يشير إلى الصف والثاني إلى العمود . مثلا

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{pmatrix}$$

عبارة عن مصفوفة ثنائية عدد عناصرها $N \times M$. x_{ij} هو عنصر الموجود في الصف الأول والعمود الأول ، x_{ij} هو العنصر الموجود في الصف الثالث والعمود الرابع ، x_{MN} هو العنصر الموجود في الصف M والعمود N . نكتبه الكتابة بلغة البيسك هي $X(I, J)$ حيث I هو المؤشر للصف ويأخذ قيم من 1 إلى M ، J هو المؤشر للعمود ويأخذ القيم من 1 إلى N . أي أن $X(2, 3)$ هو عنصر في المصفوفة X الواقع في الصف الثاني والعمود الثالث . وهكذا . بنفس الطريقة السابقة يتم حجز الخلايا المطلوبة لحزن عناصر المصفوفة ثنائية وباستخدام ايعاز DIM . وتكون الصيغة العامة بالشكل التالي :

```
N DIM NAME1(N1, M1), NAME2(N2, M2)
```

مثال

```
10 DIM X(10,12), B$(3,5), A(10)
```

قراءة المصفوفة ذات بعدين

تتم قراءة المصفوفة $A(M, N)$ باستخدام الحلقات المتداخلة وبالشكل التالي :

```
FOR I=1 TO M
FOR J=1 TO N
INPUT (READ) A(I,J)
NEXT J : NEXT I
```

تسمى هذه الطريقة للقراءة بالقراءة الصفية (ROW By ROW)

```
FOR J=1 TO N
FOR I=1 TO M
INPUT (READ) A(I,J)
NEXT I : NEXT J
```

تسمى بالقراءة العمودية (COLUMN BY COLUMN)

كتابة المصفوفة ذات بعدين

يمكن كتابة المصفوفة $A(M, N)$ صفيا بالشكل التالي :

```
FOR I=1 TO M
FOR J=1 TO N
PRINT A(I,J)
NEXT J : NEXT I
```


او عموديا بالشكل التالي :

```
FOR J=1 TO N
FOR I=1 TO M
PRINT A(I,J)
NEXT I : NEXT J
```

مثال

اكتب برنامجا لقراءة المصفوفة (10 . 5) X، اطلع عناصر المصفوفة صفيا وعلى شكل مصفوفة عدد صفوفها خمسة وعدد اعمدتها عشرة .

```
10 DIM X(5,10)
20 FOR I=1 TO 5
30 FOR J=1 TO 10
40 INPUT X(I,J)
50 NEXT J : NEXT I
60 FOR I=1 TO 5
70 FOR J=1 TO 10
80 PRINT X(I,J) ;
90 NEXT J : PRINT
100 NEXT I
```

ملاحظة

ان استخدام الرمز \circ في عبارة رقم 80 هو لطبع عناصر الصف الواحد في سطر واحد
ثم الانتقال الى سطر اخر باستخدام عبارة PRINT بعد عبارة NEXT

7.3 امثلة محلولة :

مثال (١) اذا كانت $B = (-1, -2, 3, 9, -5, 12, 122, -99)$

تمثل مصفوفة احادية ، اكتب برنامجا :

١ - لقراءة المصفوفة B

٢ - جد عدد الاعداد الموجبة وعدد الاعداد السالبة

٣ - جد عدد الاعداد الفردية وعدد الاعداد الزوجية

```
10 DIM X(8)
20 P=0:N=0:E=0:O=0
30 FOR I=1 TO 8
40 READ X(I)
50 IF X(I)>0 THEN P=P+1:GOTO 70
60 IF X(I)<0 THEN N=N+1
70 IF X(I)/2=INT(X(I)/2) THEN E=E+1:GOTO 90
80 O=O+1
90 NEXT I
100 DATA -1, -2, 3, 9, -5, 12, 122, -99
110 PRINT "P=";P,"N=";N
120 PRINT "E=";E,"O=";O
130 END
```

مثال (٢)

إذا كانت $A(50)$ مصفوفة احادية ، اكتب برنامجا لقراءة A ثم جد قيمة Y, Z حيث :

$$Y = x_1 + x_3 + \dots + x_{49}$$

$$Z = x_2 + x_4 + \dots + x_{50}$$

```
10 DIM X(50)
20 Y=0:Z=0
30 FOR I=1 TO 50
40 INPUT X(I):NEXT I
50 FOR I=1 TO 50 STEP 2
60 Y=Y+X(I)
70 Z=Z+X(I+1)
80 NEXT I
90 PRINT 'y= ';Y;' Z= ';Z
100 END
```

مثال (٣)

إذا كان $X(N)$ مصفوفة احادية عناصرها اعداد عشوائية ، اكتب برنامجا لقراءة المصفوفة ثم رتب المصفوفة ترتيبا تصاعديا .

```
5 REM Arranging an Array in Ascending Order
10 INPUT N
20 DIM X(N)
30 FOR I=1 TO N
40 INPUT X(I):NEXT I
50 FOR I=1 TO N-1
60 FOR J=I+1 TO N
70 IF X(I)<=X(J) THEN 90
```

```

80 M=X(I):X(I)=X(J):X(J)=M
90 NEXT J
100 NEXT I
110 FOR I=1 TO N
120 PRINT X(I);
130 NEXT I
140 END

```

مثال (٤)

إذا كانت A(N)$ مصفوفة عناصرها عبارة عن أسماء اشخاص ، اكتب برنامجا لترتيب هذه الاسماء ترتيبا ابجديا .

```

5 REM Arranging a list of names
10 INPUT N
20 DIM A$(N)
30 FOR I=1 TO N
40 INPUT A$(I):NEXT I
50 FOR I=1 TO N-1
60 FOR J=I+1 TO N
70 IF A$(I)<=A$(J) THEN 90
80 M$=A$(I):A$(I)=A$(J):A$(J)=M$
90 NEXT J
100 NEXT I
110 FOR I=1 TO N
120 PRINT A$(I)
130 NEXT I
140 END

```

مثال (٥)

إذا كانت :

$$B = \begin{pmatrix} -1 & 5 \\ 4 & -2 \end{pmatrix}, \quad A = \begin{pmatrix} 2 & 1 \\ 3 & 5 \end{pmatrix}$$

مصفوفتان ، اكتب برنامجاً لقراءة المصفوفتين A , B ثم جد

١ - المصفوفة C حيث : $C = A + B$

٢ - المصفوفة D حيث : $D = A * B$

واطبع المصفوفات C , D

```
10 DIM A(2,2),B(2,2),C(2,2),D(2,2)
20 FOR I=1 TO 2
30 FOR J=1 TO 2
40 READ A(I,J),B(I,J)
50 C(I,J)=A(I,J)+B(I,J)
60 NEXT J:NEXT I
70 FOR I=1 TO 2
80 FOR J=1 TO 2
90 D(I,J)=0
100 FOR K=1 TO 2
110 D(I,J)=D(I,J)+A(I,K)*B(K,J)
120 NEXT K:NEXT J:NEXT I
130 FOR I=1 TO 2
140 FOR J=1 TO 2
150 PRINT C(I,J);:NEXT J:PRINT
160 NEXT I:PRINT
170 FOR I=1 TO 2
180 FOR J=1 TO 2
190 PRINT D(I,J);:NEXT J:PRINT:NEXT I
200 DATA 2,-1,1,5,3,4,5,-2
```

مثال (٦)

إذا كانت $A(M, N)$ مصفوفة ثنائية ، اكتب برنامجاً لقراءة المصفوفة ثم جد مجموع كل سطر ثم اطبع أكبر مجموع .

```
10 INPUT M,N
20 DIM A(M,N),S(N)
30 FOR I=1 TO M
40 S(I)=0
50 FOR J=1 TO N
60 READ A(I,J)
70 S(I)=S(I)+A(I,J)
80 NEXT J
90 NEXT I
100 MAX=S(1)
110 FOR I=2 TO M
120 IF MAX > S(I) THEN 140
130 MAX=S(I)
140 NEXT I
150 PRINT 'Max = ';MAX
160 DATA 2,-1,3,5
170 END
```

مثال (7)

إذا كان A مصفوفة مربعة حيث :

$$A = \begin{pmatrix} 5 & 7 & 4 \\ 3 & 9 & 7 \\ 11 & 10 & 4 \end{pmatrix}$$

اكتب برنامجا لقراءة المصفوفة ثم نجد :

1 - مجموع العناصر الواقعة على القطر الرئيسي .

2 - اطبع الاعداد الاولى الواقعة على القطر الرئيسي وتحتها .

```
10 DIM A(3,3),P(9)
20 FOR I=1 TO 3
30 FOR J=1 TO 3
40 READ A(I,J)
50 IF I<> J THEN 70
60 S=S+A(I,J)
70 IF I<J THEN 120
80 FOR K=2 TO A(I,J)-1
90 IF A(I,J)=INT(A(I,J)/K)*K THEN 120
100 NEXT K
110 L=L+1:P(L)=A(I,J)
120 NEXT J
130 NEXT I
140 PRINT '          THE MATRIX IS '
150 FOR I=1 TO 3
160 FOR J=1 TO 3
170 PRINT A(I,J);:NEXT J:PRINT :NEXT I
180 PRINT :PRINT
190 PRINT '          THE SUM = ';S:PRINT
200 FOR I=1 TO L
210 PRINT ' P(';I;')= ';P(I)
220 NEXT I
230 DATA 5,7,4,3,9,6,11,10,4
240 END
```

THE MATRIX IS

```
5 7 4
3 9 6
11 10 4
```

THE SUM = 18

```
P( 1)= 5
P( 2)= 3
P( 3)= 11
```

تأريـن

- 1 - مصفوفة ابعادها (3*4) ، اكتب برنامجا لايجاد :
 - 1 - حاصل جمع العناصر الزوجية لكل سطر
 - 2 - حاصل ضرب العناصر الفردية لكل عمود
- 2 - اكتب برنامجا لحسب معدلات احدى الشعب اذا علمت ان كل طالب يدرس ستة مواضيع ، ثم رتب الشعبة حسب المعدل .
- 3 - اكتب برنامجا لقراءة مصفوفة احادية ثم جد واطبع مربعات ومكعبات عناصر المصفوفة .
- 4 - اكتب برنامجا لقراءة المصفوفة A حيث :

$$A = \begin{pmatrix} 3 & 0 & 1 \\ 11 & -2 & 3 \\ 5 & 7 & 14 \end{pmatrix}$$

- ثم جد واطبع
- 1 - مجموع عناصر القطر
 - 2 - عدد الاعداد الفردية منها
 - 3 - عدد الاعداد الزوجية منها
 - 4 - هذور المصفوفة A.

الفصل السادس

- البرامج الفرعية -

البرامج الفرعية THE SUBPROGRAMES

البرنامج الفرعي عبارة عن جزء من البرنامج الرئيسي والذي يمكن استدعائه اكثر من مرة واحدة وفي مواضيع مختلفة لاجراء عمليات معينة تتطلب تنفيذها لمرات عديدة وبذلك تقل عدد خطوات البرنامج الرئيسي و توفر مواضيع تخزين في وحدة الذاكرة الرئيسية .

يستفاد ايضا من البرامج الفرعية لحل المسائل الكبيرة والمعقدة وذلك بتجزئتها الى مجموعة من المسائل الصغيرة والتي ترتبط بالبرنامج الرئيسي .
تقسم البرامج الفرعية الى :

6.1 البرامج المكتبية (الدوال الفرعية) LIBRARY SUBPRGRAMME

هناك برامج تدعى برامج مكتبية مخزونة في ذاكرة كل جهاز وفي جزء خاص منها تسمى مكتبة الجهاز . كل برنامج يقوم بعملية حسابية يستدعى عند الحاجة وفي اي موقع من البرنامج الرئيسي وان سعة المكتبة وعدد البرامج المخزونة تختلف من جهاز الى اخر . من هذه الدوال :

- SQR (X) : ايجاد الجذر التربيعي لـ X
- EXP (X) : ايجاد قيمة الدالة الاسية e^x
- ABS (X) : ايجاد القيمة المطلقة للمتغير X
- SGN (X) : اشارة X الجبرية (الموجب او السالب)
- SIN (X) : ايجاد جيب الزاوية X (مقاسا بالزوايا القطرية)
- COS (X) : ايجاد جيب تمام الزاوية X (مقاسا بالزوايا القطرية)
- TAN (X) : ايجاد ظل الزاوية X
- LOG (X) : ايجاد اللوغاريتم الطبيعي لـ X
- RND (X) : ايجاد عدد عشوائي بين صفر وواحد
- INT (X) : ايجاد عدد الجزء الصحيح لعدد حقيقي

تعتبر دالة INT من الدوال المهمة في الرياضيات . تستخدم في كثير من العمليات الرياضية ومنها :

1 - تقريب عدد حقيقي الى اقرب عدد صحيح وحسب المعادلة $INT (X + 0.5)$

مثال :

اذا كان $X = 7.2$ فان $INT (X+0.5) = 7$

اذا كان $X = 7.6$ فان $INT (X+0.5) = 8$

2 - ايجاد الجزء الكسري Z من العدد الحقيقي X وحسب المعادلة :

$$Z = X - INT(X)$$

مثال

اذا كان $X = 12.321$ فان $Z = .321$

3 - تقريب الجزء الكسري لاي عدد حقيقي X الى عدد N من المراتب وحسب المعادلة

$$INT(10^N * X + .5) * 10^{-N}$$

مثال

قرب العدد $X = 5.326657$ الى ثلاث مراتب بعد الفارزة ($N = 3$)

$$(X * 10^N + .5) = 5.326657 * 10^3 + .5 = 5326.657 + .5 = 5327.157$$

$$INT(X * 10^3 + .5) * 10^{-3} = 5327 * 10^{-3} = 5.327$$

4 - إيجاد باقي القسمة لأي عددين صحيحين . إذا كان Y , X عددان صحيحان فان $Z = X - Y * INT(X / Y)$ هو باقي القسمة .

مثال

إذا كان $X = 9$, $Y = 7$ فان $7 * INT(9 / 7) = 9 - 7 = 2$

ملاحظة

يمكن استخدام قيمة Z لمعرفة ما إذا كان العدد X يقبل القسمة على Y أي مقارنة Z مع الصفر .

إذا كان $Z = 0$ فان X يقبل القسمة على Y

إذا كان $Z \neq 0$ فان X لا يقبل القسمة على Y

مثال

جد مجموع الأعداد التي تقبل القسمة على 3 والتي تقع بين (1-20)

```

10 S=0
20 FOR I=1 TO 20
30 R=I-INT(I/3)*3
40 IF R<>0 THEN 60
50 PRINT TAB(I);I;;S=S+I
60 NEXT I:PRINT:PRINT :PRINT '      S= ',S
70 END

```

6.2 الدوال الفرعية Functions

يستخدم هذا النوع من البرامج الفرعية عندما يحتاج المبرمج لتعريف عملية رياضية او مجموعة عمليات رياضية على شكل دالة ويقسم الى :

1 - الدالة ذات سطر واحد SINGLE - LINE FUNCTION

يستخدم لتعريف عملية رياضية التي تتكرر في اكثر من موقع واحد بجمللة واحدة .
وتكتب عادة في بداية او نهاية البرنامج الرئيسي وتستدعى بالاشارة الى اسمها في البرنامج . والصيغة العامة هي :

$$\text{DEF FN NAME} (V_1 \text{ و } V_2 \text{ و } \dots \text{ و } V_N) = \text{EXP.} (V_1 \text{ و } V_2 \text{ و } \dots \text{ و } V_N)$$

حيث ان DEF : كلمة البيسك الاساسية لتعريف الدالة مأخوذة من
DEFINITION

FN' : مختصر كلمة الدالة .

NAME : اسم الدالة التي تستدعى بها

$V_1 \text{ و } V_2 \text{ و } \dots \text{ و } V_N$: اسماء المتغيرات الداخلة في العملية الرياضية

$\text{EXP}(V_1 \text{ و } V_2 \text{ و } \dots \text{ و } V_N)$: هو التعبير الرياضي الذي يمثل العملية الحسابية

ملاحظة

ان بعض الحاسبات الالكترونية مثل الوركاء والتي تم تنفيذ معظم برامج هذا الكتاب عليها لاتقبل اكثر من متغير واحد في تعريف الدالة ذات سطر واحد

مثال

$$Z = \frac{X^3 + 2X^2 - X}{Y^3 + 2Y^2 - Y}$$

تتبع برنامجا لاجاد قيمة Z حيث ان

```

10 DEF FNA(N) = N^3 + 2 * N^2 - N
20 INPUT X و Y
30 LET M1=FNA(X)
40 LET M2=FNA(Y)
50 LET M=M1/M2
60 PRINT 'Z=' و M
70 END

```

مثال :

اكتب برنامجا لحساب حاصل قسمة عددين وحساب الباقي باستخدام دالة سطر واحد تحتوي على اكثر من متغير .

```

10 REM Define and Find
20 DEF FNS(I,J)=INT(I/J)
30 DEF FNH(I,J)=I-J*FNS(I,J)
40 S=FNS(45,4):H=FNH(45,4)
50 PRINT 'S=';S,, 'H=';H

```

ملاحظة

ان استخدام فارزات متتابعة في السطر رقم (50) تعني ترك حقول فارغة بحيث يكون تأثير كل فارزة حقل .

ملاحظة

يمكن استخدام دالة ذات سطر واحد لتعريف تعبير رمزي في بعض الحاسبات وليست كلها . والصيغة العامة تكون بالشكل التالي :

N DEF FN NAME\$ = 'EXPRESSION'

EXPRESSION : عبارة عن تعبير رمزي

مثال

```
10 DEF FNA$ = 'NAME'
20 LET X$ = FNA$
30 LET Y$ = FNA$
40 PRINT X$ و Y$
50 END
```

2- دالة متعدد السطور MULTI-LINE FUNCTION

تستخدم هذه الدالة لايجاد قيمة واحدة فقط نتيجة لتنفيذ مجموعة عمليات حسابية : والصيغة العامة هي :

```
N DEF FN NAME( V1 و V2 و ... )
.
.
.
M FNEND
```

اي توجد بداية للدالة وهي **DEF FN NAME**

V_1, V_2, \dots : اسماء المتغيرات الداخلة في العمليات الحسابية

FNEND : نهاية الدالة الفرعية

ان القيمة الناتجة بعد تنفيذ العمليات الحسابية سوف تخزن في المتغير الذي اسمه FN NAME بالاضافة الى كونه اسماً للدالة .

ملاحظة

ان دالة متعددة السطور غير معرفة للحاسبة الالكترونية الوركاء .

مثال

باستخدام دالة متعددة السطور اكتب برنامجا لحساب الدالة $F(X)$ حيث :

$$F(X) = \begin{cases} 2X + 1 & X > 0 \\ 1 & X = 0 \\ 2 |X| + 1 & X < 0 \end{cases}$$

```

10 INPUT X
20 DEF FNA(X)
30 IF X>0 THEN 70
40 IF X<0 THEN 90
50 FNA=1
60 GOTO 110
70 FNA=2*X+1
80 GOTO 110
90 FNA=2*ABS(X)+1
100 FNEND
110 PRINT "X", "F(X)"
120 PRINT X , FNA(X)
130 GOTO 10

```


مثال

اكتب برنامجا فرعيا لايحاجد $N!$: $(N! = 1*2* \dots *(N-1)*N)$

```
10 DEF FNX(N)
20 F=1
30 FOR I=2 TO N
40 F=F*I
50 NEXT I
60 FNX=F
70 FNEND
```

6.3 الروتينيات الفرعية SUBROUTINES

لكتابة البرنامج الفرعي هناك طريقة اخرى اكثر عموما وهي الروتينات الفرعية .
الهدف من استخدامها لايختلف عن هدف استخدام البرامج الفرعية الاخرى ،
ولكن طريقة كتابة الروتين الفرعي لاتخضع لنفس القواعد الدقيقة التي تخضع لها
تعريف البرامج الفرعية الاخرى ، حيث ان الروتين الفرعي قد يبدأ بأي امر من
اوامر لغة البيسك ولاداعي لبدا البرنامج الفرعي بامر خاص ولكن المهم ان ينتهي
الروتين الفرعي بامر هو RETURN أي العودة الى البرنامج الرئيسي والى نفس
المكان الذي توقف عنده قبل الانتقال الى الروتين الفرعي بعد الحصول على النتائج
المطلوبة من البرنامج الفرعي . يتم الانتقال الى الروتين الفرعي من البرنامج الرئيسي
باستخدام ايعاز GOSUB وتكون الصيغة للروتين الفرعي كالآتي :

```
N GOSUB M
.
.
M
.
.
L RETURN
```

ملاحظات

- 1 - يمكن للبرنامج الواحد ان يحتوي على اكثر من روتين فرعي .
- 2 - يمكن ان يضم روتين فرعي روتين فرعي اخر .
- 3 - غالبا ماتوضع عبارات الروتين الفرعي في نهاية البرنامج اي بعد كلمة END ولكن تنفيذها تتم قبل نهاية البرنامج .
- 4 - يمكن الحصول على اكثر من نتيجة واحدة عند استخدام الروتين الفرعي .
- 5 - لا يقتصر عمل الروتين الفرعي على اجراء العمليات الحسابية فقط ، بل يمكن استخدامه في العمليات المنطقية ايضا .

مثال (١)

اكتب برنامجا باستخدام روتين فرعي لاييجاد مساحة دوائر بانصاف اقطار مختلفة على التوالي : 2,4,6,8,10,12

```
5 REM Find The Area of a Circle
10 PI=22/7
20 FOR R=2 TO 12 STEP 2
30 GOSUB 70
40 PRINT ' Area =';A
50 NEXT R
60 END
70 REM SUBROUTINE STARTED
80 A=PI*R^2
90 RETURN
```

```
Area = 12.571429
Area = 50.285714
Area = 113.14286
Area = 201.14286
Area = 314.28571
Area = 452.57143
```

مثال (٢)

اكتب برنامجاً لقرئة المصفوفات $X(10)$, $Y(10)$, $Z(10)$ ثم جد قيمة V حيث :

$$V = \frac{\sum_{i=1}^{10} \sqrt{X_i^2}}{\sum_{i=1}^{10} X_i \cdot Z_i \sum_{i=1}^{10} Z_i^2}$$

١ - باستخدام دالة فرعية

٢ - باستخدام روتين فرعي

```

10 DEF FNF(A,B)
20 DIM A(10),B(10)
30 S=0
40 FOR I=1 TO 10
50 S=S+A(I)*B(I)
60 NEXT I
70 FNF=S
80 FNFEND
90 DIM X(10),Y(10),Z(10)
100 FOR I=1 TO 10
110 INPUT X(I),Y(I),Z(I):NEXT I
120 M1=FNF(X,X)
130 M2=FNF(Y,Z)
140 M3=FNF(Z,Z)
150 V=SQR(M1)/M2*M3
160 PRINT "V=";V
170 END

```

```

10 INPUT N
20 DIM X(N),Y(N),Z(N),A(N)
30 FOR I=1 TO N
40 INPUT X(I):A(I)=X(I):NEXT I
50 GOSUB 170
60 M1=S
70 FOR I=1 TO N
80 INPUT Y(I):A(I)=Y(I):NEXT I
90 GOSUB 170
100 M2=S
110 FOR I=1 TO N
120 INPUT Z(I):A(I)=Z(I):NEXT I
130 GOSUB 170
140 M3=S
150 V=SQR(M1)/M2*M3
160 PRINT 'V= ';V:END
170 REM Subroutine started
180 S=0
190 FOR I=1 TO N
200 S=S+A(I):NEXT I
210 RETURN

```

مثال (٣)

أكتب برنامجا فرعيا لايجاد حاصل ضرب مصفوفتان قابلتا للضرب ، ثم استخدم
نفس البرنامج الفرعي لايجاد حاصل ضرب B, A

```
10 GOTO 110
20 REM subroutine
30 REM DIM A(M,N),B(N,L)
40 FOR I=1 TO M
50 FOR J=1 TO L
60 C(I,J)=0
70 FOR K=1 TO N
80 C(I,J)=C(I,J)+A(I,K)*B(K,J)
90 NEXT K:NEXT J:NEXT I
100 RETURN
110 INPUT M,N,L
120 DIM A(M,N),B(N,L),C(M,L)
130 FOR I=1 TO M
140 FOR J=1 TO N
150 INPUT A(I,J):NEXT J:NEXT I
160 FOR I=1 TO N
170 FOR J=1 TO L
180 INPUT B(I,J):NEXT J:NEXT I
190 GOSUB 20
200 FOR I=1 TO M
210 FOR J=1 TO L
220 PRINT C(I,J);
230 NEXT J
240 PRINT :NEXT I
```

تمارين

1 - اكتب برنامجا لحساب :

$$R1 = \binom{N}{I} = \frac{N!}{I!(N-I)!}$$

$$R2 = \frac{N!}{(N-I)!}$$

2 - ادا كان $A(100)$, $B(100)$, اكتب برنامجا لقراءة المصفوفات A , B ثم جد معدل كل مصفوفة باستخدام روتينين فرعيين .

3 - اكتب برنامجا لانيجاد قيمة Y من المعادلة :

$$y = \sqrt[5]{x^3 + 2x + 3}$$

باستخدام دالة فرعية ، ثم اطبع قيمة Y

4 - اكتب روتيننا فرعيا لحساب

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

ثم اكتب برنامجا رئيسيا لاستدعاء الروتين الفرعي وحساب

$$Z = (x \sinh(x) - 2 \sinh(2x)) \sinh(5)$$

الفصل السابع

ايعازات أخرى في لغة البيسك
الاوامر في لغة البيسك ومفاتيح
التحكم

7.1 ايعازات اخرى في لغة البيسك

7.1.1 ايعاز RESTORE

يستخدم هذا الايعاز لاعادة قراءة القيم التي تمت تحديدها بايعاز DATA • الصيغة العامة هي : RESTORE N

مثال

```
10 READ X و Y
20 Z = Y + X
30 PRINT Z
40 RESTORE
50 READ M و N
60 PRINT M , N
70 DATA 10 و 20
80 END
```

في هذا المثال يتم تحديد قيم X , Y بواسطة DATA ، في سطر رقم 70 وبعد اجراء اللازم وباستخدام ايعاز RESTORE يتم اعطاء نفس القيم الى المتغيرات M , N وذلك بالرجوع الى عبارة رقم 70 .

7.1.2 ايعاز TAB

يستخدم هذا الايعاز لتحديد بداية موقع الطبع بالنسبة للنتائج على الوسط الخارجي (اي تحديد العمود المطلوب طبع النتائج عليه) ، ويستخدم هذا الايعاز فقط مع ايعاز PRINT .
الصيغة العامة هي TAB (X)

حيث ان X عبارة عن :

1 - ثابت عددي

2 - متغير عددي

3 - تعبير حسابي

مثال

```
10 INPUT A
20 PRINT TAB(3);A
30 END
```

في هذا المثال يتم تعريف المتغير A ومن ثم طبعه في العمود رقم 3 وذلك باستخدام ايعاز TAB

مثال

```
10 READ X$ و Y$ و Z
20 PRINT TAB (2);X$;TAB (8);Y$;TAB (18); Z
30 DATA 'HIS ' و 'AVRAGE =' و 70
```

بعد تنفيذ هذا البرنامج ستكون صيغة الطبع كالآتي :

2	8	18

HIS	AVRAGE=	70

ملاحظة

يمكن استخدام هذا الایعاز لرسم اشكال هندسية او لطبع حروف كبيرة .

مثال (١)

اكتب برنامجا لطبع حرف Z

```
10 FOR I=1 TO 5
20 PRINT TAB(I); '*';
30 NEXT I
40 PRINT
50 FOR I=4 TO 1 STEP -1
60 PRINT TAB(I); '*'
70 NEXT I
80 FOR I=1 TO 5
90 PRINT TAB(I); '*';
100 NEXT I
110 END
```

```
*****
      *
     *
    *
   *
  *
 *
*****
```

مثال (٢)

اكتب برنامجا لطبع الشكل التالي :

```
10 FOR I=1 TO 5
20 FOR J=1 TO I
30 PRINT TAB(5);"m";
40 NEXT J:PRINT:NEXT I
50 FOR I=1 TO 4
60 FOR J=5 TO I+1 STEP -1
70 PRINT TAB(5);"m";:NEXT J:PRINT :NEXT I
```

```
m
mm
mmm
mmmm
mmmmm
mmmm
mmmm
mmmm
mm
mm
m
```

مثال

اكتب برنامجا يكتب جدول تحويل الانجات الى سنتيمترات من (5 - 1) انجاً .

```
10 PRINT "Inches";TAB(8);"Centemeters"
20 PRINT "-----",TAB(8);"-----"
30 FOR N=1 TO 5
40 CM=N*2.54
50 PRINT TAB(2);N;TAB(10);CM
60 NEXT N
70 END
```

Inches	Centemeters
-----	-----
1	2.54
2	5.08
3	7.62
4	10.16
5	12.7

7.1.3 ايعاز SPC

يستخدم هذا الایعاز ایضا مع ايعاز PRINT ولكنه یحدد عدد الفراءعات بین متغیر ومتغیر اخر والصیغة هی : SPC (X)
حيث ان X اما ثابت عددي ، متغیر عددي او تعبير حسابي

مثال

```
10      ...      .....  
20  PRNIT X ; SPC(4);Y
```

7.1.4 ايعازات MID\$ ، RIGHT\$ ، LEFT\$

ايعاز LEFT\$

يقوم هذا الایعاز باخذ (سحب) عدد معين من الحروف من متغیر رمزي ومن اليسار والصیغة العامة هی :
LEFT\$(d\$ و X)
حيث ان d\$: هو المتغیر الرمزي (الغير العددي)
X : عدد الحروف المطلوبة

```
10  LET  A$ = 'ABCDEF'  
20  LET  B$ = LEFT$(A$,3)  
30  PRINT ' B$ = ' و B$  
40  END
```

مثال

في هذا المثال يتم تحديد B\$ من A\$ وذلك باخذ الحروف الثلاثة الاولى ومن اليسار ، وعند تنفيذ البرنامج تكون النتيجة كما يلي : B\$=ABC

ايعاز RIGHT\$

يقوم هذا الایعاز باخذ (سحب) عدد معين من الحروف من متغير غير عددي ومن اليمين الى اليسار والصيغة العامة هي :

RIGHT\$(d\$, X)

مثال

```
10 LET A$ = '12345'  
20 FOR I = 1 TO 5  
30 PRINT RIGHT$(A$,I) : NEXT I  
40 END
```

يتم طبع الارقام بالشكل التالي :

```
5  
4 5  
3 4 5  
2 3 4 5  
1 2 3 4 5
```

ايعاز MID\$

يقوم هذا الایعاز باخذ عدد معين من الحروف ومن موقع معين من متغير غير عددي والصيغة العامة هي :

MID\$(d\$, Y, X)

حيث d\$: متغير غير عددي

Y : الموقع المراد تحديده

X : عدد الحروف المطلوبة

7.1.5 الايغاز LOCATE

يحدد هذا الايغاز الموقع الحقيقي لكل رمز على الشاشة اي توجيه مؤشر الشاشة الى النقطة المطلوبة (X و Y) .
ان شاشة كل جهاز مقسمة الى مجموعة من الوحدات وان هذا التقسيم يختلف من جهاز الى اخر . في جهاز الوركاء الشاشة مقسمة الى مجموعة من الوحدات وان المدى لـ Y و X كلاتي :

$$0 < X \leq 39$$

$$0 < Y \leq 19$$

مثال

```
10 X$ = '125AB'
20 CLS : FOR I=1 TO 5
30 LOCATE 3+I, I-1
40 PRINT LEFT$(X$,I)
50 NEXT I : END
```

مثال

اكتب برنامجا لطبع الاعداد من 1 - 10 وعلى شكل المثلث التالي :

```
      1
     2 3
    4 5 6
   7 8 9 10
```

```
10 READ N : P = 1
20 FOR I = 1 TO N
30 LOCATE 25 - I, 10 + I
40 FOR J = 1 TO I
50 IF P > N THEN 90
60 PRINT P ; : P = P + 1
70 NEXT J : PRINT : NEXT I
80 DATA 10
90 END
```

7.2 الوامر في لغة البيسك COMMANDS IN BASIC

هناك مجموعة من الاوامر التي تعطى للحاسبة الالكترونية عن طريق لوحة المفاتيح مباشرة اثناء تشغيله وذلك لاداء اغراض مختلفة وحسب نوع الامر . تتميز الايعازات عن الاوامر بان الايعازات يجب ان تبدأ برقم السطر بينما الاوامر يجب ان تبدأ بحرف ابجدي . من هذه الاوامر :

7.2.1 الامر قائمة LIST COMMAND

يؤدي هذا الامر الى ظهور البرنامج بكامله او جزءا منه على الشاشة مما يسمح لنا بمراجعة البرنامج واجراء التعديلات والتغيرات اذا لزم الامر . والصيغة العامة هي:

LIST (لاطهار البرنامج ككله)
LIST N (لاطهار سطر الذي رقمه N)
LIST N - M (لاطهار الاسطر N الى M)

ملاحظة

عندما يكون المطلوب طبع البرنامج على الورق بواسطة الطابعة الخطية يستخدم الامر LIST.

7.2.2 الامر نفذ RUN COMMAND

باستخدام هذا الامر يتم تنفيذ البرنامج عندما يكون جاهزا للتنفيذ والحصول على النتائج المطلوبة وذلك بقراءة الايعازات التي يتضمنها البرنامج ثم تنفيذ كل من هذه الايعازات . والصيغة العامة هي : RUN

7.2.3 الامر احفظ SAVE COMMAND

بعد ادخال وتنفيذ البرنامج والتأكد من صحة البرنامج يمكن حفظ البرنامج على الاوساط المساعدة لل تخزين مثل القرص اللين ، شريط التسجيل . . . الخ تحت اسم معين باستخدام الامر احفظ « »
والصيغة العامة هي : `SAVE (CSAVE) 'NAME'`
`NAME` : الاسم المختار للبرنامج عند حفظه في جزء من الذاكرة
`SAVE` : لل تخزين على القرص اللين
`CSAVE` : لل تخزين على شريط التسجيل

7.2.4 الامر احمّل LOAD COMMAND

يمكن استرجاع البرنامج المخزون على القرص اللين (شريط التسجيل) الى ذاكرة الحاسبة باستدعاء اسم البرنامج باستخدام الامر احمّل
والصيغة العامة هي `LOAD (CLOAD) 'NAME'`
`NAME` : اسم البرنامج المراد استدعاءه الى ذاكرة الحاسبة
`LOAD` : لتحميل البرنامج المخزون على القرص اللين
`CLOAD` : لتحميل البرنامج المخزون على شريط التسجيل

7.2.5 الامر جديد NEW COMMAND

يؤدي هذا الامر الى مسح جميع البرامج المخزونة في ذاكرة الحاسبة والصيغة العامة هي : `NEW`

7.2.6 الامر احذف DELETE COMMAND

يمكن حذف سطر او مجموعة اسطر من البرنامج عند الطلب باستخدام الامر احذف . و الصيغة العامة هي :

DELETE N (N الحذف السطر الذي رقمه N)

DELETE N - M (M إلى N الحذف الاسطر من)

7.2.7 الامر قتل KILL COMMAND

يؤدي هذا الامر الى مسح البرنامج المخزون على القرص اللين والصيغة العامة

هي : KILL NAME

NAME : اسم البرنامج المراد مسحه

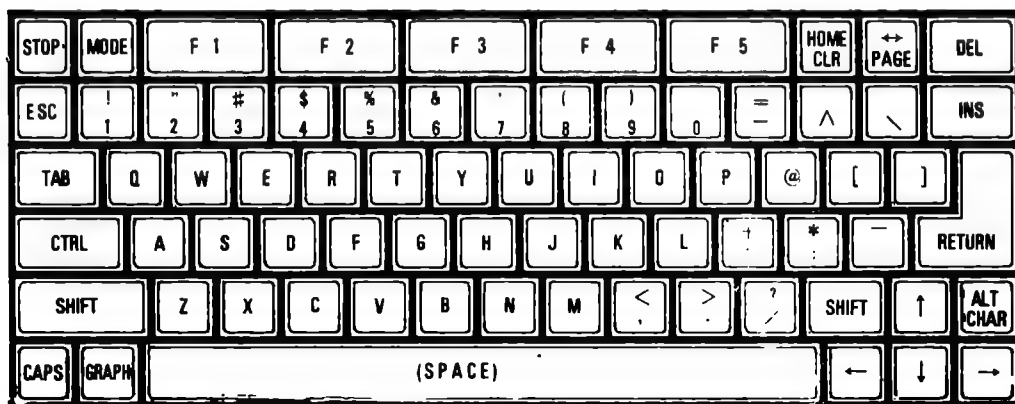
ملاحظة

هناك اوامر وايعازات اخرى في لغة البيسك يمكن الاطلاع عليها من خلال دليل كل جهاز .

لوحة المفاتيح ومفاتيح التحكم

ان لوحة المفاتيح تحتوي على مجموعة من الرموز الخاصة بالاضافة الى مجموعة من المفاتيح التي تسمى مفاتيح السيطرة .

KEYARRAY



فيما يلي اهم مفتاح نسيطة والتي نراها على لوحة المفاتيح مع وظيفة كل مفتاح :

1 - مفتاح RETURN

يستخدم لادخال سطر المكتوب الى الحاسبة وخزنها في الذاكرة ومن ثم الانتقال الى سطر جديد .

2 مفتاح SHIFT

يتم استخدام هذا المفتاح عندما يراد ادخال الرموز الموجودة في القسم العلوي من مفتاح الرموز وذلك بالضغط على هذا المفتاح ومفتاح الرمز المطلوب في ان واحد .

3 - مفاتيح الاسهم

وهي اربعة مفاتيح يحمل كل منها سهمًا يتجه الى احد لاتجاهات الاربعة (الاعلى ، الاسفل ، اليمين ، اليسار) وعند الضغط على اي منها يتحرك المؤشر على الشاشة بنفس اتجاه السهم المرسوم .

4 - مفتاح INS : من INSERT

يستخدم لادخال رموز جديدة بين رموز اي سطر .

5-مفتاح DEL : من DELETE

يستخدم لمسح رمز واحد او اكثر من رموز السطر .

6 - مفتاح HOME , CLR أو HOME /CLEAR

يستخدم هذا المفتاح لتنظيف الشاشة كليًا وعندها يقف المؤشر في الزاوية العليا اليسرى من الشاشة .

هناك مفاتيح اخرى على لوحة مفاتيح كل جهاز حيث يقوم كل مفتاح بعملية معينة ، تختلف هذه المفاتيح من جهاز الى جهاز اخر .

ملاحظة

يمكن تعديل البرنامج اما باضافة سطر جديد وادخاله بعد اعطائه رقما مناسبًا ثم الضغط على مفتاح (RETURN) . ان الرقم الذي يعطى للسطر يجب ان يكون واقعا بين رقمي السطرين اللذين يضاف بينهما السطر الجديد ، او حذف سطر معين

وذلك بادخال رقم ذلك السطر ثم الضغط على مفتاح (RETURN) كذلك
يمكن تعديل سطر معين في البرنامج بتوجيه المؤشر على ذلك السطر وإلى الموقع
المطلوب تعديله ثم الضغط على مفتاح (RETURN) .

تمارين

1 - اكتب برنامجاً لطبع الحروف A ، B ، C ، D وبالشكل التالي :

```

A
A B
A B C
A B C D

```

تم ادراج الحروف التالية باستخدام ايجاد TAB

```

M  ٠ ٠ ٠ ٠ ٠
n  ٠ ٠ ٠ ٠ ٠
٧  ٠ ٠ ٠ ٠ ٠

```

```

      .
     .
    .
   .
  .
 .
.

```

الفصل الثامن

امثلة متنوعة في الرياضيات

مثال (١)

اكتب برنامجا لتحويل الزاوية مقدرة بالتقدير الزاوى الى تقدير نصف قطرى .

```
5 PRINT "This program transfers from degree into radia
10 PI=22/7
20 INPUT "input the degree";
30 Y=X*PI/180:PRINT "Y=";
40 END
```

مثال (٢)

اكتب برنامجا لحساب N من حدود متسلسلة فيبوناشي . متسلسلة فيبوناشي (Fibo- nacci) عبارة عن اعداد صحيحة بحيث ان كل عدد يساوى مجموع العددين السابقين له .

```
10 L=1 : K=0
20 INPUT N
30 FOR I=1 TO N
40 M=L+K:L=K:K=M
50 PRINT M :NEXT I
```

مثال (٣)

اوجد قيمة المجموع التالي :
$$\sum_{i=1}^3 \sum_{j=2}^4 \sum_{k=3}^5 (i + j + k)$$

```
10 S=0
20 FOR I=1 TO 3
30 FOR J=2 TO 4
40 FOR K=3 TO 5
50 S=S+I+J+K
55 NEXT K:NEXT J:NEXT I
60 PRINT "S=";S
70 END
```

مثال (٤)

اكتب برنامجاً لإيجاد المضاعف المشترك الأصغر (Least Common Multiple)
بين العددين L , K .

```
10 INPUT L,K
20 J=1
30 M=L*K
40 IF M=INT(M/K)*K THEN 60
50 J=J+1 :GOTO 30
60 PRINT 'The Multiple least common between 'L' and 'K' is 'M
```

مثال (٥)

اكتب برنامجاً للحصول على N من الأرقام العشوائية الصحيحة ، قيمها بين الصفر و
(10)

```
10 INPUT N
20 FOR L=1 TO N
30 Z=10*RND(1)
40 I=INT(Z)
50 PRINT I: NEXT L
60 END
```

مثال (٦)

اكتب برنامجاً لتحويل العدد X من أي نظام ذات أساس B إلى النظام العشري .

```
10 INPUT 'INPUT THE NUMBER AS A STRING AND THE BASE B';A$,B
20 N=LEN(A$):S=0
30 FOR I=1 TO N
40 M$=MID$(A$,I,1)
50 K=VAL(M$)
60 S=S+K*B^(I-1):NEXT I
70 PRINT 'THE NUMBER IN DECIMAL =';S
80 END
```

مثال (٧)

اكتب برنامجا لقراءة الأزواج المرتبة الثمانية واطبع الأزواج التي حاصل جمع أعضائها يقبل القسمة على خمسة . علما بان الأزواج المرتبة هي :

(9,1), (12,3), (5,5), (5,1), (0,3), (2,7), (2,3), (7,1)

```
10 FOR I=1 TO 8
20 READ X,Y :S=X+Y
30 R=S-5*INT(S/5)
40 IF R<>0 THEN 60
50 PRINT '(';X;',';Y;')'
60 NEXT I
70 DATA 9,1,12,3,5,5,5,1,0,3,2,7,2,3,7,1
80 END
```

مثال (٨)

اكتب برنامجا لرمي زهر النرد عدد من المرات ، وفي كل مرة يجب ان نعرف ما هو الرقم العشوائي الناتج من رمي زهر النرد .

```
10 INPUT 'INPUT ANY NUMBER';N
20 FOR I=1 TO N
30 R=INT(6*RND(1))+1:PRINT
50 INPUT 'INPUT THE EXPECTED RANDOM NUMBER';X
60 IF X=R THEN 80
70 PRINT 'NO,the Right Number Is';R :PRINT
75 GOTO 90
80 PRINT 'You Are Right'
90 NEXT I
100 END
```

مثال (٩)

اكتب برنامجا لتحويل القراءات التالية :

187 , 155 , 0 , 78 , 9823 , 45

من السنتيمتر الى متر وستمتر .

```

5 REM Relation between Centemeter & Meter
10 FOR I=1 TO 6
20 READ L
30 IF L=0 GOTO 60
40 M=INT(L/100);C=L-INT(L/100)*100
50 PRINT L;" = ";M;" Met.  AND ";C;" CENT."
60 NEXT I
70 DATA 187,155,0,78,9823,45

```

مثال (١٠)

باستخدام ايعاز ON-GOTO ، اكتب برنامجا لايجاد قيمة Z المعروفة بالشكل .

$$Z = \begin{cases} (a+2bx+dx^2) \sin x & \text{if } x = -1 \\ x + 2 & \text{if } x = 0 \\ |e^x| & \text{if } x = 1 \end{cases}$$

```

10 INPUT A,B,D
20 INPUT X
30 ON X+2 GOTO 40,50,60
40 Z=(A+2*B*X+D*X^2)*SIN(X):GOTO 70
50 Z=X+2:GOTO 70
60 Z=ABS(EXP(X))
70 PRINT "When X=";X;"      Z=";Z
80 GOTO 20
90 END

```


مثال (١١)

اكتب برنامجا لايجاد جدول الضرب للاعداد من (0 , 9) ثم اطبع الجدول

```
10 DIM A(11,11)
20 FOR I=0 TO 9
30 FOR J=0 TO 9
40 IF I=0 THEN A(I,J)=J:GOTO 70
50 IF J=0 THEN A(I,J)=I:GOTO 70
60 A(I,J)=I*J
70 PRINT A(I,J);TAB(3*J+1)
80 NEXT J:PRINT
90 NEXT I
```

0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

مثال (١٢)

اكتب برنامجا بلغة البيسك لايجاد المجموع التالي :

$$S = \sum_{m=1}^3 \sum_{k=1}^3 \frac{k! m!}{(k+m)!}$$

```

10 REM FIND DOUBLE SUM
20 FOR M=1 TO 3
30 FOR K=1 TO 3
40 N=K:GOSUB 100:A=R
50 N=M:GOSUB 100:B=R
60 N=K+M:GOSUB 100:C=R
70 S=S+A*B/C:NEXT K:NEXT M
80 PRINT "S=";S
90 END
100 REM SUBROUTINE STARTED
110 R=1:FOR I=2 TO N
120 R=R*I:NEXT I
130 RETURN

```

مثال (١٣)

اكتب برنامجا لتحويل العدد N من النظام العشري اي نظام اخر

```

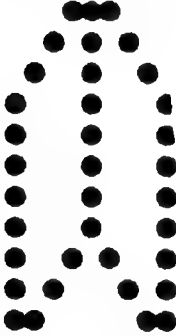
10 PRINT " This program transfers "from decimal "
15 PRINT " into any other system"
20 DIM A$(1000)
30 INPUT "input the number to be transferred ";N
40 INPUT "input the base";B
50 X=INT(N/B):P=N-X*B
60 J=J+1
70 FOR I=0 TO B-1
80 IF R=I THEN A$(J)=STR$(I):GOTO 100
90 NEXT I
100 IF X=0 THEN 120
110 N=X:GOTO 50
120 FOR I=J TO 1 STEP -1
130 PRINT A$(I):NEXT I
140 END

```

مثال (١٤)

اكتب برنامجاً لطبع الشكل التالي :

```
10 FOR I=1 TO 4
20 LPRINT TAB(4-I);"●";TAB(4);"●";
30 LPRINT TAB(I+4);"●":NEXT I
40 FOR J=1 TO 4
50 LPRINT "●";
60 LPRINT TAB(4);"●";
65 LPRINT TAB(8);"●":NEXT J
70 FOR I=1 TO 3
75 LPRINT "●";
80 LPRINT TAB(4-I);"●";
90 LPRINT TAB(I+4);"●";
100 LPRINT TAB(8);"●":NEXT I
```



مثال (١٥)

اكتب برنامجاً يقوم بقراءة مجموعة من الدرجات المئوية والفهرنهايتية وحسب ما مؤشر
ازاء كل درجة ويقوم بتحويل الدرجات المئوية الى الفهرنهايتية وبالعكس ،
باستخدام العلاقات التالية :

$$\text{Celsius} = 5/9 (F - 32)$$

$$\text{Fahrenheit} = (9/5 * C) + 30$$

اذ علمت ان الدرجات المعطاة هي :

120F , 55F , 42C , 30C , 80F , -350C , 152F , 100F
0F , -32F , -24C , 360C

```

10 FOR I=1 TO 10
20 READ T,T$
30 IF T$='F' GOTO 70
40 F=(9/5*T)+32
50 PRINT T,'IN Cel.=' ,F,'In feh.'
60 GOTO 90
70 C=5/9*(T-32)
80 PRINT T,'In feh.=' ,C,'In cel.'
90 NEXT I
100 DATA 80,'F',30,'C',42,'C',55,'F'
110 DATA 120,'F',360,'C',-24,'F',-32,'F'
120 DATA 0,'F',100,'F',152,'F',-350,'C'

```

80	In feh.=	26.666667	In cel.
30	IN Cel.=	86	In feh.
42	IN Cel.=	107.6	In feh.
55	In feh.=	12.777778	In cel.
120	In feh.=	48.888889	In cel.
360	IN Cel.=	680	In feh.
-24	In feh.=	-31.111111	In cel.
-32	In feh.=	-35.555556	In cel.
0	In feh.=	-17.777778	In cel.
100	In feh.=	37.777778	In cel.

مثال (١٦)

أكتب برنامجاً لقراءة المصفوفة $A(N)$ حيث ان
 $N = 9$ و $A = (5, 6, 7, 9, 0, 34, 2, 21, 11)$ ثم حول المصفوفة الى مصفوفة ثنائية
 $B(M, L)$ بحيث
 $(M * L = N)$
ثم اطبع المصفوفة الثنائية $B(M, L)$

```
10 READ N,M,L
20 DIM A(N),B(M,L)
30 FOR I=1 TO N
40 READ A(I):NEXT I
50 PRINT 'The One Dimentional Array is':PRINT
60 FOR I=1 TO N
70 PRINT A(I);:NEXT I:PRINT
80 P=1
90 FOR I=1 TO M
100 FOR J=1 TO L
110 B(I,J)=A(P)
120 P=P+1
130 NEXT J:NEXT I:PRINT
140 PRINT 'The Two Dimentional Array is':PRINT
150 FOR I=1 TO M
160 FOR J=1 TO L
170 PRINT B(I,J);:NEXT J
180 PRINT :NEXT I
190 DATA 9,3,3
200 DATA 5,6,7,9,0,34,2,21,11
```

The One Dimentional Array is

5 6 7 9 0 34 2 21 11

The Two Dimentional Array is

5 6 7
9 0 34
2 21 11

مثال (١٧)

اكتب برنامجا لاجاد قيمة الخطا في عملية البتر عند حساب قيمة الدالة :

$$Y = X - \sin X$$

عند النقطة $X = \frac{\pi}{6}$ باستخدام متسلسلة تايلر .

```
10 PRINT "This Program Calculates The Truncation Error"
20 PRINT
30 DEF FNF(X)=1-COS(X)
40 INPUT N
50 S=0:X=22/28
60 FOR I=2 TO N STEP 2
70 F=1
80 FOR J=2 TO I
90 F=F*J:NEXT J
100 S=S+(-1)^((I+2)/2)*X^I/F
110 NEXT I
120 EX=FNF(X)
130 EP=S
140 ER=EX-EP
145 PRINT "The Exact Value= ";EX
150 PRINT "The Aproximate Value= ";EP
160 PRINT "The Error Value= ";ER
```

This Program Calculates The Truncation Error

```
The Exact Value= .29311679
The Aproximate Value= .29279358
The Error Value= 3.2320188E-04
```

مثال (١٨)

أوجد قيمة تقريبية للجذر 5 (اي ان $X^2 - 5 = 0$) بحيث ان نسبة الخطأ اقل من 10^{-6} ، بطريقة نيوتن - رافسن ، علماً بان القيمة التقريبية للجذر 2.5 ، ثم قرب الناتج الى خمسة ارقام عشرية واحسب عدد التكرارات للحصول على الجذر المطلوب .

ملاحظة

طريقة نيوتن - رافسن هي :

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)} , n = 0, 1, 2, \dots$$

X_n : هي قيمة تقريبية اوليه

X_{n+1} : هي قيمة تقريبية افضل من X_n

$f(X_n)$: قيمة الدالة عند $X=X_n$

$f'(X_n)$: قيمة مشتقة الدالة. عند $X=X_n$

```

10 REM FIND THE ROOT BY NEWTON RAPHSON METHOD
20 DEF FNF(X)=X^2-5
30 DEF FNG(X)=2*X
40 X0=2.0:I=0
50 F1=FNF(X0):F2=FNG(X0)
60 X1=X0-F1/F2
70 IF(X1-X0)<1E-08 THEN 90
80 X0=X1:I=I+1 :GOTO 50
90 X1=X1*10^5:Y=INT(X1):Y=Y*10^(-5)
100 PRINT "The Root=";Y;" With Error<.000001"
110 PRINT "The Number of Iterations=";I
120 END

```

The Root= 2.23611 With Error<.000001
The Number of Iterations=

مثال (١٩)

إذا كان A , B متجهان ومعرفان كالآتي :

$$A = (a_1 \quad a_2 \quad a_3) = (5 \quad 3 \quad 2)$$

$$B = (b_1 \quad b_2 \quad b_3) = (4 \quad 1 \quad 1)$$

اكتب برنامجا لحساب وطبع :
١ - حاصل ضرب القياسي $A \cdot B$ حيث : $A \cdot B = \sum_{i=1}^3 a_i b_i$

٢ - طول كل من المتجهين $A \cdot B$
 $|A| = \sqrt{a_i^2}$

٣ - جيب تمام الزاوية المحصورة بين المتجهين .

$$\cos \theta = \frac{A \cdot B}{|A| \cdot |B|}$$

```

10 REM A,B VECTORS
20 READ N
30 DIM A(N),B(N)
40 FOR I=1 TO N
50 S1=0:S2=0:S3=0
55 READ A(I),B(I)
60 S1=S1+A(I)^(B(I))
70 S2=S2+A(I)^2
80 S3=S3+A(I)^3
90 NEXT I
100 LA=SQR(S2)
110 LB=SQR(S3)
120 T=S1/(LA*LB)
130 PRINT "THE DOT PRODUCT=";S1
140 PRINT "THE LENGTH OF A=";LA
145 PRINT "THE LENGTH OF B=";LB
150 PRINT "THE ANGLE BETWEEN A AND B=";T
160 DATA 3
170 DATA 5,4,3,1,2,1
180 END

```


مثال (٢٠)

اكتب برنامجا لقراءة مصفوفة ثنائية $A(M, N)$ ثم حول المصفوفة الثنائية إلى مصفوفة احادية $B(L)$ حيث $(L = M * N)$ واطبع المصفوفة الاحادية .

```
10 READ M,N :L=M*N
20 DIM A(M,N),B(L)
30 P=1
40 FOR I=1 TO M
50 FOR J=1 TO N
60 READ A(I,J)
70 B(P)=A(I,J)
80 P=P+1
90 NEXT J:NEXT I
100 PRINT 'The Two Dimentional Array is ':PRINT
110 FOR I=1 TO M
120 FOR J=1 TO N
130 PRINT A(I,J);:NEXT J:PRINT:NEXT I:PRINT
140 PRINT 'The One Dimentional Array is':PRINT
150 FOR I=1 TO L
160 PRINT B(I);:NEXT I
170 DATA 3,3
180 DATA 4,5,7,2,9,0,2,8,8
190 END
```

The Two Dimentional Array is

```
4 5 7
2 9 0
2 8 8
```

The One Dimentional Array is

```
4 5 7 2 9 0 2 8 8
```

مثال (٢١)

إذا كان لدينا ثلاث أحرف A , B , C اكتب برنامجاً لطبع جميع الترتيبات الممكنة لهذه الحروف .

```
10 PRINT 'Arrang any three Diffrent Letters in Three Diffrent Ways'

15 PRINT
20 DIM X$(3)
30 FOR I=1 TO 3
40 READ X$(I):NEXT I
50 N=1
60 FOR I=2 TO 3
70 N=N*I:NEXT I

80 PRINT 'The Number of Diffrent Ways to be Arranged=';N:PRINT

90 FOR K1=1 TO 3
100 FOR K2=1 TO 3
110 IF K1=K2 THEN 140
120 LET K3=6-(K1+K2)
130 PRINT TAB(5);X$(K1);X$(K2);X$(K3)
140 NEXT K2
150 NEXT K1
160 DATA 'A','B','C':END
```

Arrang any three Diffrent Letters in Three Diffrent Ways'

The Number of Diffrent Ways to be Arranged= 6

ABC
ACB
BAC
BCA
CAB
CBA

ملاحظة

نلاحظ عند إيجاد قيمة K_3 ، ان مجموع الأرقام الصحيحة الثلاثة K_1 , K_2 , K_3 يساوي ستة (6) لذلك :

$$K_3 = 6 - (K_1 + K_2)$$

وبطريقة اخرى :

```
10 PRINT 'Arrang any three Diffrent Letters in Three Diffrent Ways'

15 PRINT
20 DIM X$(3)
30 FOR I=1 TO 3
40 READ X$(I):NEXT I
50 N=1
60 FOR I=2 TO 3
70 N=N*I:NEXT I

80 PRINT 'The Number of Diffrent Ways to be Arranged=';N:PRINT

90 FOR I=1 TO 3
100 FOR J=1 TO 3
110 IF J=I THEN 170
120 FOR K=1 TO 3
130 IF K=J OR K=I THEN 160
150 PRINT TAB(5);X$(I);X$(J);X$(K)
160 NEXT K
170 NEXT J
180 NEXT I
200 DATA 'A','B','C':END
```

Arrang any three Diffrent Letters in Three Diffrent Ways

The Number of Diffrent Ways to be Arranged= 6

```
ABC
ACB
BAC
BCA
CAB
CBA
```

مثال (٢٢)

اكتب برنامجا لايجاد القاسم المشترك الاعظم
(Greatest Common Divisor) بين عددين .

```
5 REM find the greatest common divisor
10 DIM A(2),B(2,100),C(2),T(2),P(2,50)
15 DIM X(2,100)
20 MUL=1
30 FOR I=1 TO 2
35 PRINT
40 INPUT A(I):Y=A(I):GOSUB 300:PRINT
50 C(I)=Y
60 FOR K=1 TO C(I)
70 B(I,K)=X(I,K)
80 NEXT K
100 K=1:M=0
110 IF K=C(I) THEN 135
120 IF K>C(I) THEN 160
130 IF B(I,K)=B(I,K+1) THEN K=K+1: GOTO 110
135 M=M+1
140 P(I,M)=B(I,K):K=K+1
150 GOTO 110
160 T(I)=M
170 NEXT I
180 FOR I=1 TO T(1)
190 FOR J=1 TO T(2)
200 IF P(1,I)<>P(2,J) THEN 230
210 MUL=MUL*P(1,I)
220 GOTO 240
230 NEXT J
240 NEXT I
245 PRINT
250 PRINT "The gcd=";MUL
260 PRINT
290 END
300 REM subroutine started
310 V=1
320 FOR J=2 TO Y
```

```

330 R=Y-INT(Y/J)*J
340 IF R<>0 THEN 390
350 X(I,V)=J:PRINT X(I,V);TAB(3);
360 Y=INT(Y/X(I,V))
370 IF Y=1 THEN 400
380 V=V+1:GOTO 320
390 NEXT J
400 RETURN

```

مثال (٢٣)

إذا كان A مصفوفة $N \times N$ اكتب برنامجاً لإيجاد محدد A ومن ثم إيجاد معكوس A ، باستخدام طريقته معامل المرافق .
تتلخص هذه الطريقة كالآتي : -

إذا كان $A(N \times N)$ مصفوفة مربعة و $|A| \neq 0$ ، فإن مصفوفة المعاملات المرافقة لـ A هي مصفوفة $B(A)$ مربعة $N \times N$ حيث أبدل كل عنصر فيها بمعاملها المرافق . والمصفوفة المجاورة للمصفوفة A هي مدور مصفوفة المعاملات المرافقة ولتكن $C(B)$ إذن فإن :

$$A^{-1} = \frac{C(B)}{|A|}$$

```

10 READ N
20 DIM A(N,N):DIM B(N-1,N-1):DIM C(N,N):DIM M(N,N)
30 FOR I=1 TO N
40 FOR J=1 TO N
50 READ A(I,J)
60 NEXT J
70 NEXT I
73 PRINT "          The Matrix is ":PRINT
75 FOR I=1 TO N
77 FOR J=1 TO N
78 PRINT A(I,J);:NEXT J:PRINT :NEXT I
80 FOR P=1 TO N
90 FOR L=1 TO N

```

```

100 FOR I=1 TO N
110 FOR J=1 TO N
120 IF P<I AND L<J THEN B(I-1,J-1)=A(I,J): GOTO 160
130 IF P<I AND L>J THEN B(I-1,J)=A(I,J): GOTO 160
140 IF P>I AND L<J THEN B(I,J-1)=A(I,J): GOTO 160
150 IF P>I AND L>J THEN B(I,J)=A(I,J)
160 NEXT J:NEXT I
170 FOR K=1 TO N-1
180 IF B(K,K)=0 THEN 260
190 FOR I=1 TO N-1
200 FOR J=N-1 TO 1 STEP -1
210 R=I+K:T=J+K-1
220 IF R>N-1 THEN 250
230 IF T>N-1 THEN 250
240 B(R,T)=(-B(R,K)/B(K,K)*B(K,T))+B(R,T)
250 NEXT J:NEXT I:NEXT K:GOTO 340
260 FOR X=K TO N-1
270 IF B(X,K)=0 THEN 310
280 FOR F=K TO N-1
290 B(K,F)=B(X,F)+B(K,F)
300 NEXT F:GOTO 190
310 NEXT X
320 NEXT K
340 Y=1
350 FOR I=1 TO N-1
360 Y=B(I,I)*Y
370 NEXT I
380 C(P,L)=(-1)^(P+L)*Y
384 NEXT L:NEXT P
385 Z=0
390 FOR J=1 TO N
400 Z=Z+A(1,J)*C(1,J)

405 NEXT J
410 PRINT "The Value of the determent is ";Z:PRINT
420 IF Z=0 THEN PRINT "This Matrix has not have invers":END
430 FOR I=1 TO N
440 FOR J=1 TO N
450 M(I,J)=(1/Z)*C(I,J)
460 NEXT J:NEXT I
465 PRINT :PRINT "      THE INVERS MATRIX IS":PRINT

```

```

470 FOR I=1 TO N
480 FOR J=1 TO N
490 PRINT M(J,I);
500 NEXT J:PRINT
510 NEXT I
520 DATA 2,1,0,-1,1

```

The Matrix is

```

2 0 1
1 5 6
5 9 1

```

The Value of the determent is =-114

THE INVERS MATRIX IS

```

.42982456-.078947368 .043859649
-.25438596 .026315789 .096491228
.14035088 .15789474-.087719298

```

مثال (٢٤)

اكتب برنامجا لايجاد معكوس المصفوفة A ، وذلك باستخدام الصيغة التالية :

اذا كان $A(N*N)$ مصفوفة مربعة وان $P(N*N)$ ايضا مصفوفة مربعة وان $AP = I$ فان $A^{-1} = P$. اي ان المطلوب هو ايجاد المصفوفة P

```

10 REM FIND THE INVERSE OF A MATRIX
15 REM USING THE IDENTITY MATRIX
17 READ N
20 DIM A(2*N,2*N)
30 FOR I=1 TO N
40 FOR J=1 TO N
50 READ A(I,J)
60 NEXT J:NEXT I
70 REM PUT IDENTITY MATRIX

```

```

80 NI=2*N
90 FOR I=1 TO N
100 FOR J=N+1 TO NI
110 A(I,J)=0
120 NEXT J :NEXT I
130 FOR I=1 TO N
140 A(I,I+N)=1:NEXT I
150 REM THE MATRIX AND COULOM VECTOR'
160 PRINT
170 PRINT 'THE MATRIX      '
180 PRINT '-----'
190 FOR I=1 TO N
200 FOR J=1 TO NI
210 PRINT A(I,J);
220 NEXT J:PRINT:NEXT I
230 REM FIND INVERS MATRIX
240 FOR I=1 TO N
250 P=A(I,I)
260 IF P=0 THEN GOSUB 440
270 FOR J=1 TO NI
280 A(I,J)=A(I,J)/P
290 NEXT J
300 FOR K=1 TO N
310 IF I=K THEN 340
315 P=A(K,I):FOR L=1 TO NI
320 A(K,L)=A(K,L)-A(I,L)*P
330 NEXT L
340 NEXT K
345 NEXT I:PRINT
350 PRINT '    INVERES    MATRIX'
360 PRINT '-----'
370 FOR I=1 TO N
380 FOR J=N+1 TO NI
390 PRINT A(I,J);:NEXT J
400 PRINT
410 NEXT I
415 DATA 2
420 DATA 1,0,-1,1
430 END

```



```

440 REM SUBROUTINE STARTED
700 FOR K=I+1 TO N
710 FOR J=K-1 TO NI
720 IF A(K,J)=0 THEN 800
730 V=A(I,J):A(I,J)=A(K,J)
732 A(K,J)=V
740 NEXT J
800 NEXT K
805 W=B(I):B(I)=B(I+1):B(I+1)=W
810 P=A(I,I)
820 RETURN

```

THE MATRIX

```

-----
1 0 1 0
-1 1 0 1

```

INVERES MATRIX

```

-----
1 0
1 1

```

مثال (٢٥)

اكتب برنامجا لحل مجموعة معادلات من الدرجة $N \times N$ بطريقة معكوس المصفوفة .

إذا كان $AX = B$ فإن $X = A^{-1} B$

حيث A هي مصفوفة المعاملات (معاملات المجاميل) :

$$A = \begin{bmatrix} a_{11} & a_{12} \dots a_{1N} \\ a_{21} & a_{22} & a_{2N} \\ \vdots & \vdots & \vdots \\ a_{M1} & a_{M2} & a_{MN} \end{bmatrix}$$

وان $X = (x_1, x_2, \dots, x_n)$ عمود المجاهيل و
 $B = (b_1, b_2, \dots, b_m)$ عمود الكميات المطلقة .

```

10 DIM A(10,20),B(10),X(10)
20 READ N
30 FOR I=1 TO N
40 FOR J=1 TO N
50 READ A(I,J)
60 NEXT J:NEXT I
70 FOR I=1 TO N
80 READ B(I)
90 NEXT I
100 REM PUT DIAGONAL MATRIX
110 NI=2*N
120 FOR I=1 TO N
130 FOR J=N+1 TO NI
140 A(I,J)=0
150 NEXT J :NEXT I
160 FOR I=1 TO N
170 A(I,I+N)=1:NEXT I
180 REM THE MATRIX AND COULOM VECTOR"
190 PRINT
200 PRINT 'A MATRIX          B(I)'
210 PRINT '-----'
220 FOR I=1 TO N
230 FOR J=1 TO NI
240 PRINT A(I,J);
250 NEXT J:PRINT TAB(20);B(I)
260 PRINT
270 NEXT I
280 REM FIND INVERS MATRIX
290 FOR I=1 TO N
300 P=A(I,I)
302 PRINT'PA=';P
305 IF P=0 THEN GOSUB 700
310 FOR J=1 TO NI
320 A(I,J)=A(I,J)/P
330 NEXT J
340 FOR K=1 TO N
350 IF I=K THEN 400
360 P=A(K,I)

```

```

370 FOR L=1 TO NI
380 A(K,L)=A(K,L)-A(I,L)*P
390 NEXT L
400 NEXT K
405 FOR J=1 TO N
406 FOR M=1 TO NI

407 PRINT A(J,M);:NEXT M:PRINT TAB(20);B(J):PRINT:NEXT J

410 NEXT I
420 PRINT "    INVERES    MATRIX"
430 PRINT "-----"

440 FOR I=1 TO N
450 FOR J=N+1 TO NI
460 PRINT A(I,J);:NEXT J
470 PRINT
480 NEXT I
490 FOR I=1 TO N
500 X(I)=0
510 FOR J=N+1 TO NI
520 X(I)=X(I)+A(I,J)*B(J-N)
530 NEXT J
540 NEXT I
550 PRINT
555 PRINT "THE VALUE OF THE VARIABLES"
560 PRINT "-----"
570 FOR I=1 TO N
580 PRINT "X(";I;")=";X(I)
590 PRINT
600 NEXT I
610 DATA 3
620 DATA 1,1,2,0,1,-3,0,0,5,1,-3,10
630 END
700 PRINT "I=";I:FOR K=I+1 TO N
710 FOR J=K-1 TO NI
720 IF A(K,J)=0 THEN 800
730 V=A(I,J):A(I,J)=A(K,J)
732 A(K,J)=V
740 NEXT J
800 NEXT K
805 W=B(I):B(I)=B(I+1):B(I+1)=W
810 P=A(I,I):PRINT "P=";P
820 RETURN

```

مثال (٢٦)

اكتب برنامجا في الاحصاء لقراءة مجموعتان $X(N)$, $Y(N)$ ، ثم ايجاد :

- ١ - المعدل لكل مجموعة .
- ٢ - معامل الارتباط .
- ٣ - الانحراف القياسي .
- ٤ - معامل الانحراف .
- ٥ - معادلة الانحدار .

```

360 PRINT ' ';TAB(10);' ';TAB(25);' '
370 FOR I=1 TO N
380 PRINT I;TAB(10);X(I);TAB(25);Y(I)
390 NEXT I

400 INPUT'DO YOU HAVE TO CHANGE ANY DATA PUT Y FOR YES OR N FOR NO';A#

410 IF A#='Y'OR A#='Y' THEN 430
420 GOTO 500
430 INPUT'PUT THE NO OF ROW TO BE CHANGED ';F
440 IF F>N THEN 430
450 INPUT 'THE VALUE OF X AND Y';X(F),Y(F)
460 INPUT'DO YOU HAVE ANY OTHER CHANGE PUT(Y) OR (Y)';A#
470 IF A#='Y' OR A#='Y' THEN 430
480 GOTO 345
500 SX=0:SY=0:XY=0:XQ=0:YQ=0:DX=0:DY=0
510 FOR I=1 TO N
520 SX=SX+X(I)
530 SY=SY+Y(I)
540 XY=XY+X(I)*Y(I)
550 XQ=XQ+X(I)*Y(I)
560 YQ=YQ+Y(I)*Y(I)
570 NEXT I

580 REM      AX:AVRAGE OF X
590 REM      AY:AVRAGE OF Y
600 AX=SX/N;AY=SY/N
620 REM      FIND CORRELATION OF COFF.
630 R1=XY-N*AX*AY
640 R2=SQR((XQ-N*AX^2)*(YQ-N*AY^2))
650 R=R1/R2
660 REM      DY=STANDARED DVIATION OF X
670 REM      DX=STANDARED DVIATION OF Y
675 DX=SQR(XQ-N*AX*AX)/(N)
680 DY=SQR(YQ-N*AY*AY)/(N)
690 REM      CV=COFFICIENT OF
700 CV=DY/AY*100
710 B=(XY-(SX*SY/N))/(XQ-(SX^2/N))
720 A=AY-B*AX
730 REM      PRINT THE RESULTS
810 SCREEN 3,2,2:CLS
812 COLOR 3
820 LINE (0,0)-(319,190),4,B

```

```

830 FOR I=1 TO 20:PRINT'*';:NEXT I
840 FOR I=1 TO 18:LOCATE 0,I:PRINT'*':NEXT I
845 FOR I=0 TO 19
850 LOCATE I,18:PRINT'*';:NEXT I
860 FOR I=1 TO 17:LOCATE 19,I:PRINT '*':NEXT I
954 COLOR 5
955 LOCATE 2,2:PRINT'AX=';AX
960 LOCATE 2,3:PRINT'AY=';AY
970 LOCATE 2,4:PRINT'R=';R
980 LOCATE 2,5:PRINT'DX=';DX
990 LOCATE 2,6:PRINT'DY=';DY
1000 LOCATE 2,7:PRINT'CV=';CV
1005 LOCATE 2,9:PRINT'EQU.OF LIN.REG IS '
1008 LOCATE 2,12
1010 PRINT 'Y=';A; '+';B'X'
1100 DATA 25,35,10,44,25,56,25,10,35,35
1110 DATA 7,17,25,34,10,28,45,41,7,1
1112 FOR I=1 TO 1000:NEXT I

```

المصادر

- 1 - مقدمة في علم الحاسبات الالكترونية والبرمجة بلغة البيسك
تأليف : د . محمد الفيومي 1983 .
- 2 - البرمجة بلغة البيسك
تأليف : د . مهدي فاضل موسى 1989
- 3 - مبادئ علم الحاسبات الالكترونية والبرمجة بلغة البيسك
تأليف : محمد علي شلال - عبدالله الديوجي 1987 .
- 4 - الحاسبة الالكترونية وبرمجتها بلغة البيسك
ترجمة : هيثم الوردي 1985 .
- 5 - لغة البرمجة - بيسك
تأليف : سامر عبدالجبار المطليبي 1987